DECnet–RSX
System Manager's Guide
Order No. AA-H224C-TC

November 1983

This manual describes the concepts functions, and utilities related to DECnet-RSX system management

SUPERSESSION/UPDATE INFORMATION: This is a revised manual.

OPERATING SYSTEM AND VERSION: RSX-11M V4.1
RSX-11M-PLUS V2.1
RSX-11S V4.1

SOFTWARE VERSION: DECnet-11M V4.0
DECnet-11M-PLUS V2.0
DECnet-11S V4.0

To order additional copies of this document, contact your local Digital Equipment Corporation Sales Office.

digital equipment corporation · maynard, massachusetts

First Printing, June 1982
Revised, November 1983

The postage-prepaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist us in preparing future documentation.

Distributed and Mid-Range Systems Publications typeset this manual using DIGITAL's TMS-11 Text Management System.

Contents

Page

Preface

Part I Tutorial

Chapter 1 Network Management Overview

Chapter 2 Network Management Components

Chapter 3 Operating DECnet-RSX/PSI Nodes

Part II CFE, NCP, and VNP

Chapter 6 Network Management Utilities (CFE, NCP, VNP)

Part III Network Management Tools

Chapter 7 Console Carrier Requester (CCR)

Chapter 8 Host Task Loader (HLD)

Chapter 9 KMX Dump Analyzer (KDA)

Chapter 10 Network Dump Analyzer (NDA)

Chapter 11 Network Display Program (NTD)

Chapter 12 Queue Manager (QUE)

Chapter 13 Trace Interpreter Task (TRI)

Part IV Error Messages

Chapter 14 Utility Error Messages

Part V Appendixes

Appendix A CFE, NCP, and VNP Command Summary

Appendix B Object Type Codes

Appendix C Network Management Event Logger Interface

Appendix D Event Class and Type Summary

Appendix E Network Counter Summary

Appendix F Network Parameter and Counter Type Numbers

Figures

Tables

Preface

The *DECnet-RSX System Manager's Guide* presents information needed to manage a DECnet-RSX node within a DECnet network. The term DECnet-RSX collectively refers to three DECnet products:

- DECnet-11M, which runs on RSX-11M
- DECnet-11M-PLUS, which runs on RSX-11M-PLUS
- DECnet-11S, which runs on RSX-11S

This manual also describes system management procedures for RSX-11 PSI and DECnet-RSX/PSI. The term DECnet-RSX/PSI refers to an RSX system that runs RSX-11 PSI and DECnet-RSX simultaneously. Both DECnet-RSX and RSX-11 PSI products conform to the Digital Network Architecture (DNA). DNA is the model for all DECnet implementations and the standard that allows different Digital operating systems to participate in the same network. The DNA model contains the Comite Consultatif International Telegraphique et Telephonique (CCITT) recommendation X.25, which defines a standard interface from a computer or terminal to a packet-switching network (PSN). The RSX-11 PSI product implements the CCITT recommendation X.25 to enable an RSX operating system to participate in a PSN.

The utilities described in this manual are used to manage DECnet-11M V4.0, DECnet-11S V4.0, and DECnet-11M-PLUS V2.0 nodes. You can use these utilities to manage both local and remote DECnet systems, which may be other than RSX based. Although Phase IV DECnet provides for cross-system network command operation, there are also some system-specific commands. RSX system-specific commands are flagged in Appendix A. For implementations other than DECnet-RSX, refer to the system manager's guide for that system.

Intended Audience

This manual is intended for anyone who is responsible for building, maintaining, and managing the network. In this manual, all such people are collectively referred to as the system manager.

Structure of This Manual

This manual is tab divided into five distinct parts. The first part is strictly tutorial in nature and serves to point the user to the specific reference material found in the other four parts. The parts and their contents are summarized below.

Part I — Tutorial

Chapter 1 — Network Management Overview

Provides an overview of network management as it relates specifically to DECnet-RSX and briefly introduces the various network management tools and utilities.

Chapter 2 — Network Management Components

Describes the basic DECnet-RSX components and summarizes the CFE, NCP, and VNP commands and parameters relevant to each.

Chapter 3 — Operating DECnet-RSX/PSI Nodes

Describes how to start up and shut down a local DECnet-RSX or DECnet-RSX/PSI node. Also includes information on how to monitor network activity (using display commands and event-logging messages) and how to test network operation (using loopback tests and the trace facility).

Chapter 4 — Supporting Remote RSX-11S Systems

Describes down-line loading and up-line dumping of RSX-11S systems and down-line task loading for RSX-11S.

Chapter 5 — Choosing Network Buffer Parameters

Describes the allocation and usage of the various system buffers.

Part II — CFE, NCP, and VNP

Chapter 6 — Network Management Utilities (CFE, NCP, VNP)

Describes the format, parameters, and usage of all CFE, NCP, and VNP commands.

Part III — Network Management Tools

Each of the chapters in Part III describes the operation of a network management tool, including commands for its operation and examples of its use and/or output (where applicable).

Chapter 7 — Console Carrier Requester (CCR)

Chapter 8 — Host Task Loader (HLD)

Chapter 9 — KMX Dump Analyzer (KDA)

Chapter 10 — Network Dump Analyzer (NDA)

Chapter 11 — Network Display Program (NTD)

Chapter 12 — Queue Manager (QUE)

Chapter 13 — Trace Interpreter Task (TRI)

Part IV — Error Messages

Chapter 14 — Utility Error Messages

Summarizes all error messages for the following tools and utilities: CCR, CFE, HLD, NCP, NDA, NTD, NTINIT, NTL, TRI, VNP.

Part V — Appendixes

Appendix A — CFE, NCP, and VNP Command Summary

Summarizes all CFE, NCP, and VNP commands and their parameters, and identifies those that are RSX system specific.

Appendix B — Object Type Codes

Lists all valid object type codes with their process types.

Appendix C — Network Management Event Logger Interface

Describes the network management logging monitor interface that you can use to enable a user-written program to process network events.

Appendix D — Event Class and Type Summary

Describes the event message format and lists all event messages by class and type, including a brief description of each.

Appendix E — Network Counter Summary

Lists all counters maintained by the network for lines, circuits, nodes, modules, and the system, and gives a brief description of each.

Appendix F — Network Parameter and Counter Type Numbers

Lists all network parameter and counter type numbers recognized by DECnet-RSX network management software.

Associated Documents

Before reading this manual, you should have a working knowledge of DECnet and the RSX-11 operating system you are using. A prerequisite to the effective use of this manual is familiarity with the overall character of DECnet as described in the following manuals:

Introduction to DECnet (order no. AA-J055D-TC)

Overview of DECnet-RSX (order no. AA-M096B-TC)

Programming and user utility information (including PSI programming facilities) concerning DECnet-RSX is contained in the following manuals:

DECnet-RSX Guide to User Utilities (order no. AA-H223C-TC)

DECnet-RSX Programmer's Reference Manual (order no. AA-M098B-TC)

RSX-11 PSI User's Guide (order no. AA-M371A-TC)

Network generation and postinstallation checkout procedures are described in the following manuals:

DECnet-RSX Network Generation and Installation Guide (order no. AA-H225C-TC)

DECnet-RSX Postinstallation Checkout Procedures (order no. AA-J319C-TK)

RSX-11 PSI Generation Guide (order no. AA-M372A-TC)

Documentation for other Digital Ethernet Communications Server (DECS) products includes:

Introduction to Local Area Networks (order no. EB-22714-18)

Networks: Ethernet Products and Services Catalog (order no. ED-25484-42)

DECnet Router Server Installation and Operation Guide (order no. AA-X019A-TK)

DECnet Router/X.25 Gateway Installation and Operation Manual (order no. AA-Y666A-TK)

Terminal Server Installation and Operation Guide (order no. AA-X020A-TK)

Other manuals that contain information related to the material covered in this manual are:

RSX-11M System Generation and Installation Guide (order no. AA-H625B-TC)

RSX-11M Guide to Writing an I/O Driver (order no. AA-2600E-TC)

RSX-11M/M-PLUS Task Builder Manual (order no. AA-L680A-TC)

RSX-11M/M-PLUS System Management Guide (order no. AA-L679A-TC)

RSX-11M/M-PLUS Batch and Queue Operations Manual (order no. AA-L671A-TC)

RSX-11M/M-PLUS Utilities Manual (order no. AA-L681A-TC)

RSX-11 PSI System Manager's Guide (order no. AA-M370A-TC)

DECnet-VAX System Manager's Guide (order no. AA-H803C-TE)

Terminals and Communications Handbook (order no. EB-20752-20)

Information on using PSI for a specific subscription service is contained in the *RSX-11 PSI Network-specific Information* manual (order no. AA-M373A-TC).

An overview of the Digital Network Architecture (DNA) is provided in the following manual:

DECnet Digital Network Architecture (Phase IV) General Description (order no. AA-N149A-TC)

The following functional specifications provide detailed descriptions of Phase IV DNA protocols:

DNA Network Management Functional Specification, Version 4.0.0 (order no. AA-X437A-TK)

DNA Network Services Protocol Functional Specification, Version 4.0.0 (order no. AA-X439A-TK)

DNA Digital Data Communications Message Protocol (DDCMP) Functional Specification, Version 4.1.0 (order no. AA-K175A-TK)

DNA Data Access Protocol (DAP) Functional Specification, Version 5.6.0 (order no. AA-K177A-TK)

DNA Maintenance Operations Functional Specification, Version 3.0.0 (order no. AA-X436A-TK)

DNA Session Control Functional Specification, Version 1.0.0 (order no. AA-K182A-TK)

The Ethernet, A Local Area Network, Data Link Layer and Physical Layer Specifications, Version 2.0.0 (order no. AA-K759B-TK)

DNA Ethernet Node Product Architecture Specification, Version 1.0.0 (order no. AA-X440A-TK)

DNA Ethernet Data Link Functional Specification, Version 1.0.0 (order no. AA-Y298A-TK)

DNA Routing Layer Functional Specification, Version 2.0.0 (order no. AA-X435A-TK)

NOTE

The *DNA Routing Layer Functional Specification* supercedes the Phase III *Transport Functional Specification*. In the *DECnet-RSX System Manager's Guide*, the Routing layer is sometimes referred to as the Transport layer.

Acronyms

The following acronyms for DECnet-RSX and PSI components are used in this manual:

ACK	Positive acknowledge message
BCUG	Bilateral closed user group
CCB	Communication control buffer
CCITT	Comite Consultatif International Telegraphique et Telephonique
CCR	Console carrier requester
CCS	Console carrier server
CDA	Crash dump analyzer
CEX	Communications Executive
CFE	Configuration File Editor
CSR	Control status register
CUG	Closed user group
DAP	Data Access Protocol
DCB	Device control block
DCE	Data circuit-terminating equipment
DDCMP	Digital Data Communications Message Protocol
DDM	Device driver module
DECS	Digital Ethernet Communications Server
DLC	Data link control process
DLL	Down-line system loader
DLM	Data link mapping
DLX	Direct Line Access Controller
DNA	Digital Network Architecture
DSR	Dynamic storage region
DTE	Data terminal equipment
DTR	DECnet test receiver
DUK	Dump KMX task
DUM	Up-line system dumper
ECL	End Communication layer

EVL	Event Logger
EVR	Event-logging receiver
FAL	File Access Listener
FRMR	Frame reject error
FTQ	File transfer queue manager
FTS	File Transfer Spooler
HLD	Host Task Loader utility
ICB	Interrupt control block
KDA	KMX dump analyzer
KRB	Controller request block
LAPB	Link access procedure, Version B (CCITT recommendation for frame level protocol)
LDB	Large data buffer
LLC	Logical link control process
LUN	Logical unit number
MIR	Loopback mirror
MOP	Maintenance Operation Protocol
NAK	Negative acknowledge message
NCP	Network Control Program
NDA	Network Dump Analyzer
NFT	Network File Transfer utility
NICE	Network Information and Control Exchange Protocol
N(R)	Next expected sequence number
NS	DECnet user interface pseudodevice
NTD	Network Display Program
NTDEMO	Network display server
NTINIT	Network Initializer
NTL	Network loader
NW	PSI user interface pseudodevice
ODT	On-line debugging tool
PIP	Peripheral Interchange Program

PLI	PSI packet level interface
PSI	Packetnet System Interface
PSN	Packet-switching network
PVC	Permanent virtual circuit
QUE	Queue manager
RAM	Random access memory
RDB	Receive data buffer
RNR	Receive not ready
SCB	Status control block
SDB	Small data buffer
SLD	Satellite task loader
SVC	Switched virtual circuit
TKB	RSX task builder
TRI	Trace interpreter task
UCB	Unit control block
UFD	User file directory
UIC	User identification code
UMR	UNIBUS mapping register
URB	User request block
VNP	Virtual Network Processor
XDT	Executive debugging tool
XPT	Transport layer

Graphic Conventions

The following conventions are used throughout this manual.

DOT MATRIX TYPE	is used for examples of system input or output.
RED INK	is used in examples to indicate user input. Black type represents system prompts or displays.
shaded text	in command formats (Chapter 6 and Appendix A only) flags commands and parameters that are valid for PSI users only (including DLM).
UPPERCASE	represents actual characters that you must enter in commands.

NOTE

Uppercase words can be abbreviated to the first 3 or more unique characters.

lowercase italic	indicates variables whose value you must specify.
brackets [ ]	specify that the enclosed data is optional. All text not enclosed by brackets is required. For most commands in Chapter 6, you must choose at least one of the parameter options (see Chapter 6 for details). Do not type the brackets when you enter a command.
braces { }	indicate that you must choose one (and only one) of the enclosed options. Do not type the braces when you enter a command.
numbers	are decimal unless otherwise noted.
CTRL/x	means that you must press the CTRL key and the specified character key simultaneously.
RET	means that you should press the RETURN key. Even when it is not shown in examples, you should always press RETURN to enter a command line.
ESC	means that you should press the ESC key.
LF	means that you should press the line feed key.

Example:

$$\text{CLEAR} \left\{ \begin{array}{l} \text{LINE } \textit{line-id} \\ \text{KNOWN LINES} \end{array} \right\} \left\{ \begin{array}{l} \text{ALL} \\ \textbf{COUNTER TIMER} \end{array} \right\}$$

When issuing this command, you must include a specific line ID (for example, DMC-0) or specify the KNOWN LINES keyword. Then you must specify either ALL or COUNTER TIMER (the latter is valid for PSI users only). You can also abbreviate the words, as shown in the following sample command:

```
NCP>CLE LIN DMC-0 ALL RET
```

PART I
TUTORIAL

The chapters in this section of the manual contain overview discussions of network management concepts and components and introductory descriptions of material that is described in technical detail in later sections.

- Chapter 1 — Provides an overview of network management as it relates specifically to DECnet-RSX and briefly introduces the various network management tools and utilities.
- Chapter 2 — Describes the basic DECnet-RSX components and summarizes the CFE, NCP, and VNP commands and parameters relevant to each.
- Chapter 3 — Describes how to start up and shut down a local DECnet-RSX or DECnet-RSX/PSI node. Also includes information on how to monitor network activity and how to test network operation.
- Chapter 4 — Describes down-line loading and up-line dumping of RSX-11S systems and down-line task loading for RSX-11S.
- Chapter 5 — Describes the allocation and usage of the various system buffers.

Chapter 1
Network Management Overview

This chapter provides a brief overview of DECnet-RSX, with an emphasis on topics that pertain to managing local and remote DECnet nodes. This overview describes the kinds of network configurations in which DECnet-RSX nodes can participate and introduces the data bases and utilities you use to control a node's performance.

1.1 DECnet Interface with RSX Operating Systems

DECnet is the collective name for the software and hardware products that allow various Digital operating systems to participate in a network. DECnet-RSX is the software implementation that enables an RSX-11M, RSX-11M-PLUS, or RSX-11S operating system to function as a network node. As the RSX network interface, DECnet-RSX supports both the protocols necessary for communicating over the network and the functions necessary for configuring, controlling, and monitoring nodes. A DECnet-RSX node can communicate with other DECnet-RSX nodes in the network or with any other Digital operating system that supports DECnet.

A DECnet network is decentralized; nodes connected to the network can communicate with each other without having to go through a central node. There is no hierarchy. Instead, all nodes are "peers" and thus do not depend on the availability of a node on a higher level. Each node can communicate with any other node in the network to gain access to its resources.

1.2 RSX-11 Packetnet System Interface (PSI)

RSX-11 PSI is the software product that allows you to participate in a packet-switching environment. A packet-switching network (PSN) consists of switching nodes (network interfaces called data circuit-terminating equipment or DCEs) connected by links. The computers or terminals attached to the PSN are called data terminal equipment (DTE) and use Digital's implementation of the X.25 protocol to communicate with the PSN.

RSX-11 PSI can be used both for direct communication with other X.25 DTEs (the X.25 equivalent of a node) and for linking DECnet nodes across a PSN. The latter capability, where an X.25 virtual circuit is used as a DECnet data link, is known as data link mapping (DLM). In addition, a computer or terminal can be attached directly to an X.25 PSN without using DECnet.

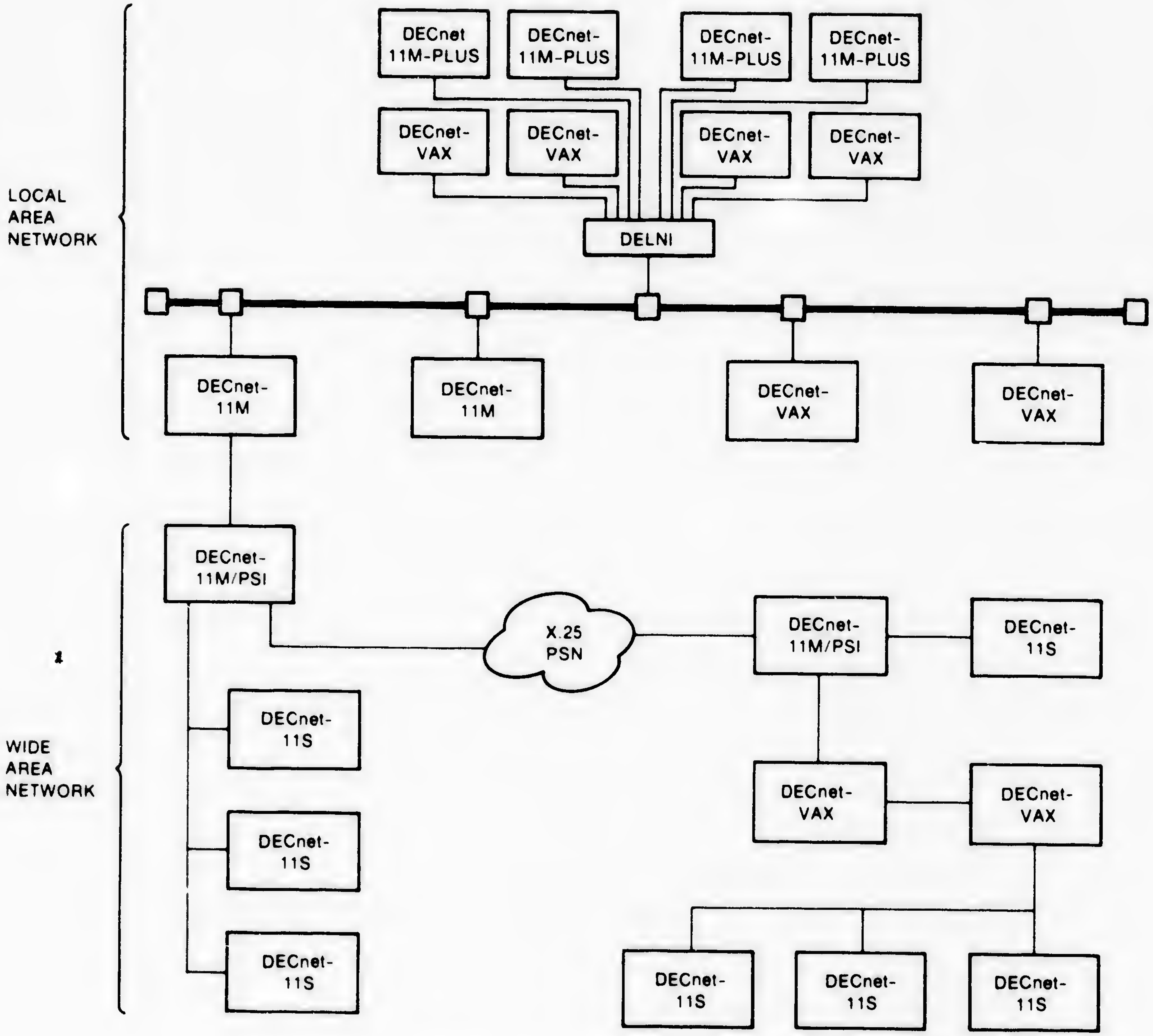


Figure 1–1: Phase IV Configuration

1.3 DECnet Configurations

DECnet-RSX supports connections to both local area and wide area networks:

- Local area network connections are provided by the Ethernet and its compatible hardware.
- Wide area network connections are provided by various means, including DDCMP point-to-point and multipoint devices and data link mapping to a PSN.

Figure 1-1 is a diagram of a typical Phase IV configuration that includes DECnet-RSX and DECnet-VAX nodes on an Ethernet, as well as DDCMP multipoint connections and data link mapping via an X.25 PSN to remote DECnet nodes. As the figure shows, a DECnet-RSX node can participate in a wide area network and a local area network simultaneously. See the *Introduction to DECnet* for more information about DECnet configurations.

1.4 Managing the Network

As system manager of a DECnet-RSX network, you have a number of key responsibilities, which include:

- At NETGEN, defining network components and their parameters in a central data base at the local node
- Configuring your node to ensure proper network operation in accordance with other nodes in the network (This includes defining node addresses, node names, buffer sizes, and circuit costs, all of which parameters affect the routing topology of the network.)
- Controlling and monitoring local and remote network operation
- Testing network hardware and software operation
- Loading systems down line to remote nodes

If your node includes RSX-11 PSI, you have the following responsibilities:

- Defining RSX-11 PSI components and their parameters in the network configuration data base and thus configuring RSX-11 PSI
- Monitoring the operation of RSX-11 PSI
- Analyzing hardware and software operation and diagnosing problems

1.5 DECnet–RSX Data Bases and Utilities

Network data bases provide the information that a DECnet-RSX node needs to function as part of a network. DECnet-RSX data bases include:

- The permanent data base
- The volatile data base
- The system image file

You use a different utility to modify or define each data base:

- The Configuration File Editor (CFE) modifies the permanent data base.
- The Network Control Program (NCP) modifies the volatile data base.
- The Virtual Network Processor (VNP) modifies the system image file. (DECnet-RSX-11M-PLUS does not support VNP.)

Chapter 6 describes the command syntax for the three utilities and provides individual command descriptions in alphabetical order. The utility or utilities to which each command applies appears at the top of the page. (Many NCP and VNP commands have the same format and definition and are therefore described together.) Error messages for CFE, NCP, and VNP, as well as other network management tools (see Section 1.7), are described in Chapter 14.

1.5.1 The Permanent Data Base and the CFE Utility

The permanent data base is the configuration file CETAB.MAC, which is a product of NETGEN. CETAB.MAC contains the data base used to load the configuration defined at NETGEN. Parameters in the permanent data base are used by DECnet-RSX software whenever you load the Communications Executive, lines, circuits, and the X.25/X.29 modules (for systems with a PSI capability). The copy of CETAB.MAC in memory after you have loaded DECnet-RSX is the volatile data base. The CFE utility modifies only the CETAB.MAC permanent data base. When you use CFE, the modifications to the generated system do not come into effect until the next time you load the affected part of the network software.

1.5.2 The System Image File and the VNP Utility

The system image file is the RSX-11 operating system that is loaded from disk into memory when the system is booted. You use VNP, a privileged utility, to add the DECnet-RSX software to the disk system image file or to modify the DECnet-RSX software already present there. (RSX-11M-PLUS does not support VNP.) VNP is comparable to the Virtual Console Monitor Routine (VMR), the utility you use to modify the RSX-11M or RSX-11S operating system image.

VNP does not change the network software running in main memory. VNP changes that you make to the system image file on disk come into effect the next time the operating system is rebooted.

1.5.3 The Volatile Data Base and the NCP Utility

The volatile data base is the DECnet-RSX system image currently in memory, and NCP is the utility you use to modify it. When you bring the system up, the volatile data base includes information from the CETAB.MAC file. Changes you make using NCP remain in effect until the system goes down or until you reload DECnet-RSX or reboot the operating system. NCP commands allow you to load, control, and monitor the running DECnet-RSX configuration as well as to test the network hardware and software and to down-line load an RSX-11S system image. If you have used VNP to load DECnet-RSX into the system image file on disk, you automatically load DECnet when you boot the system. Otherwise, you use NCP commands to load the network software after you boot the operating system.

There are two versions of NCP: the full complement supported by DECnet-11M and DECnet-11M-PLUS and a subset of those commands for DECnet-11S. Most NCP commands can be executed both at your local system and at remote DECnet systems. Certain NCP commands can be executed only by privileged users. Appendix A summarizes both command sets, specifies which commands are for privileged users only, and flags commands that cannot be executed remotely. Chapter 6 provides detailed descriptions of all NCP commands and their parameters.

1.6 Where NETGEN Ends and System Management Begins

The network generation procedure produces a DECnet-RSX system configuration with or without a PSI capability. The system includes a configuration file, command files, and network task images. The configuration file is the CETAB.MAC file, which contains information collected during network generation. The command files include the NETINS.CMD file, which is used to load DECnet-RSX; the NETCFE.CMD file, which stores the parameters selected during network generation; and the NETREM.CMD file, which is used to unload the network and remove network tasks. The network task images are those images selected for your system configuration. Given these products of the generation procedure, Figure 1-2 provides a step-by-step walk-through of the operations performed to produce a running DECnet-RSX system.

STEP 1 Generating a DECnet-RSX Configuration	STEP 2 Loading DECnet Software	STEP 3 Managing Local and Remote Nodes
Perform a network generation (**NETGEN**) NETGEN output: • CETAB.MAC • Command files: NETINS.CMD NETREM.CMD NETCFE.CMD • Network task images If necessary, issue **CFE** commands to modify CETAB.MAC, the permanent data base.	Load DECnet using one of three options: • Invoke **NETINS.CMD** (installs network tasks; optionally loads DECnet from CETAB.MAC and turns on local node); or • Issue **NCP** commands to load DECnet from CETAB.MAC and turn on local node; or • Issue **VNP** commands to load DECnet from CETAB.MAC into system image file; then reboot system.	Issue **NCP** commands to: • Modify the volatile data base • Test network software • Load remote systems • Perform other management tasks Issue **CFE** commands to modify CETAB.MAC before reloading DECnet. Issue **VNP** commands to modify the system image before rebooting the system.

Figure 1-2: Producing a Running DECnet-RSX System

Though the procedures can be iterated differently within each set, the following is a sequential description of the steps:

1. **Network generation.** Network generation provides the components upon which step 2, loading the network, operates. The *DECnet-RSX Network Generation and Installation Guide* describes the network generation procedure for configuring a DECnet-RSX system with or without RSX-11 PSI.

   You can issue CFE commands to modify CETAB.MAC (the configuration file) once it has been created by the network generation procedure. This is useful for modifying configuration parameters prior to loading the network.

2. **Loading the network.** Once you have configured the system, you can load the network software into the target configuration (that is, the actual running configuration produced by the NETGEN procedure) by invoking the NETINS.CMD command file or by using NCP or VNP commands (see Section 3.1.1). The end product of this step is an active DECnet-RSX configuration capable of participating in a network. At this point, you should run the postinstallation checkout procedures, as defined in the *DECnet-RSX Postinstallation Checkout Procedures* manual.

3. **Reconfiguring and tuning the running network.** Once you have loaded the network software, you can use NCP to modify the volatile data base to enhance network performance and reconfigure the running network (for example, turn circuits on or off and enable or disable PSI network operations). You can also use NCP to test and monitor network software operation.

1.7 Other Network Management Tools

In addition to CFE, VNP, and NCP, DECnet-RSX supports other network management tools to perform specific functions:

- Console carrier requestor (CCR)
- Host Task Loader (HLD)
- Network Dump Analyzer (NDA)
- Network Display Program (NTD)
- Queue Manager utility (QUE)
- Trace interpreter task (TRI)

Each of these tools is introduced briefly in the following sections and is described in detail in Chapters 7 through 13. Error messages associated with these utilities are described in Chapter 14.

1.7.1 Console Carrier Requestor (CCR)

The console carrier requestor (CCR) provides remote access to console carrier services on a remote Digital Ethernet Communication Server (DECS) system. CCR uses the DLX interface to communicate with the console carrier server (CCS) at the remote node. Chapter 7 describes how to run CCR.

1.7.2 Host Task Loader (HLD)

The Host Task Loader (HLD) is a utility that enables you to down-line load tasks to a target RSX-11S system that is running DECnet. This enables the target system to use a host node for storing task images (both overlaid and nonoverlaid) on disk and for providing disk space for checkpointing tasks. Chapter 4 describes the prerequisites for down-line loading tasks. Chapter 8 discusses the HLD mapping table used by the utility to handle requests from the remote (also referred to as satellite) node.

1.7.3 Network Dump Analyzer (NDA)

The Network Dump Analyzer (NDA) is a DECnet utility used to analyze crash dumps of RSX systems that were running DECnet when the crash occurred. NDA is a nonprivileged task for use only on DECnet-RSX (including those systems with a PSI capability). NDA uses the network symbol table file to compile and analyze information from a crashed system image. The utility creates both a formatted file from the dump media and a listing printout. NDA is useful only if you are familiar with internal data structures and data bases for DECnet-RSX and RSX-11 PSI.

You cannot use NDA to analyze a crash dump unless the executive crash dump routine has been built into the RSX operating system during system generation (see the *RSX-11M System Generation and Installation Guide*). Chapter 10 discusses NDA capabilities and provides a summary of the NDA command syntax and switches.

1.7.4 Network Display Program (NTD)

NTD provides real-time displays of network status information such as active network tasks and resources allocated to the system or a summary of reachable nodes. The display information changes as the status changes. You can run NTD to display resource information about your node, any other Phase IV DECnet-RSX node, or any Phase III DECnet-RSX or DECnet-IAS node in the network. Chapter 11 describes the commands used to manipulate the NTD displays and describes the display formats.

1.7.5 Queue Manager Utility (QUE)

The Queue Manager utility (QUE), supported by RSX-11M V4.0 and RSX-11M-PLUS V2.0, provides the interface to the DECnet file transfer queue. (The QUE utility itself is also a feature of the RSX operating system.) You use QUE, a privileged utility, to initialize the File Transfer Spooler (FTS) queue and processor and to manipulate user-queued FTS requests. Chapter 12 describes how to use QUE.

1.7.6 Trace Interpreter Task (TRI)

The trace interpreter task (TRI) is an RSX-11 PSI facility that allows you to diagnose software problems on an X.25 communications line. You can trace line messages and write them to a file. Use TRI to analyze and print information in this file. TRI is useful only if you are familiar with the X.25 level 2 and 3 protocols. Chapter 13 provides a summary of the TRI command syntax and switches.

Chapter 2
Network Management Components

This chapter outlines DECnet-RSX network management components and refers you to related CFE, NCP, and VNP commands and parameters that you can use to configure your system. It is assumed that you are familiar with the characteristics of DECnet as described in the *Introduction to DECnet* and the *Overview of DECnet-RSX*.

The components described in this chapter can be modified in three different data bases, depending on the network management utility command that is used, as outlined below:

Command Function	CFE (permanent data base)	NCP (volatile data base)	VNP (system image file)
Creates/modifies parameters	DEFINE	SET	SET
Clears parameters	PURGE	CLEAR	CLEAR
Displays component data	LIST	SHOW	SHOW

There is a common set of commands and parameters that apply to all DECnet systems. In addition, each DECnet system has commands and parameters that are specific to it; for example, DECnet-RSX has system-specific commands to define aliases, objects, and processes. Appendix A summarizes all CFE, NCP, and VNP commands and uses red ink to identify those that are RSX system specific.

The components that are covered in this chapter are listed below with references to the sections in which they are discussed. Where applicable, descriptions include the component ID format, a table of the network management utility parameters associated with the component and any other information that is useful in configuring the component within the network.

Component	Section
Nodes (includes access control, aliases, Ethernet addressing)	Section 2.1
Routing	Section 2.2
Objects	Section 2.3
Lines	Section 2.4
Circuits	Section 2.5
Logging	Section 2.6
Counters	Section 2.7
Processes	Section 2.8
PSI modules X.25 protocol module X.25 server module X.29 server module X.25 access module	Section 2.9

System component parameters for network buffers are discussed in Chapter 5.

2.1 Nodes

Nodes are Digital systems that run DECnet software that provides the interface between the local operating system and the network.

Three terms are applied to nodes, reflecting your relationship to the node being described:

- **Local node.** Your node — the node from which you operate.
- **Remote node.** Any node other than yours in the network.
- **Executor node.** The node at which your current network management command executes. Usually the executor is the local node. However, you can execute most NCP commands at a remote node, as described in Chapter 6. In such a case, your local node would be considered a remote node by the executor.

Many functions that the system manager performs require the identification of a specific node. For example, you can use a remote node name with the Network Display Program (NTD) to display active network information for that node. During network generation, you define your node's name and address as the executor node. You can also define node names and addresses for remote nodes. The information that you define for the executor is used in the operation of DECnet at that node.

When you configure a network at the local node, you must have node name and address entries in the volatile data base for the local node and for all nodes with which you may want to establish a logical link. The following sections describe node identification and the relevant node parameters for establishing an operational network node data base.

2.1.1 Node Identifica n

A node ID can be specified in two forms:

- **Node address.** A unique integer in the range of 1 to 1023.
- **Node name.** A unique alphanumeric character string containing 1 to 6 characters, including at least 1 alphabetic character.

To satisfy routing requirements, each node in the network must have a unique address, which is defined at network generation. Each node must also have a unique name associated with its address in order for a logical link to be established with it. If no node name has been associated with a node address at network generation, you can do so later using a SET/DEFINE NODE command. Note that the node name is known only to the local node network software, while the node address is known networkwide by the routing function.

Once you have specified both a node name and a node address, you can use either one wherever you need to specify a node ID in CFE, NCP, and VNP commands. The local DECnet-RSX software translates node names into node addresses.

To simplify remote node identification, you can assign alias node names (see Section 2.1.1.2) to destination nodes. Alias node names are known only to the local node. The network software maps alias node names to the network destination node name and address.

When you wish to execute a network management command for the executor node, you can often use the keyword EXECUTOR (abbreviated to EXE) as a short cut to specifying NODE *node-id*. For example, if you were executing commands on node BOSTON, the following two commands would result in the same display:

```
NCP SHOW NODE BOSTON
NCP SHOW EXE
```

To remove the association of a node name with a node address in the volatile data base, use the CLEAR NODE NAME command. Thereafter, your node will not recognize that node name and can no longer establish logical links to that node. To remove a node from the permanent data base, you must use the PURGE NODE ALL command.

The PURGE, LIST, and SHOW NODE commands allow you to specify a command for all known nodes, instead of for just one specific node. The KNOWN NODES keyword refers to all nodes that have been defined in the permanent data base (and — for SHOW KNOWN NODES — in the volatile data base). The SHOW NODE command also allows other more restricted plural specifications for nodes (see the discussion of SHOW commands in Chapter 3).

Loop nodes. A loop node is a special node that designates a loop path without regard for standard network routing procedures. A loop node name is associated with a specific circuit coming from the executor node. Loop nodes are used primarily for testing network hardware (see the LOOP commands in Chapter 6).

2.1.1.1 Access Control — Certain NCP commands require you to specify access control information with the node ID when you are attempting to access a remote node. This section describes the access control format and briefly describes how to set up default access control information using alias node names.

The system manager can control user access to the local system at the node or object level for inbound logical link connections. Incoming call handling for access to local DTE and PSI objects is regulated differently (see the *RSX-11 PSI System Manager's Guide*).

Access control information can be supplied in either of two formats: the standard DECnet format or an RSX-specific format.

Standard DECnet format:

node-id [USER *user-id*]
[PASSWORD *password*]
[ACCOUNT *account*]

RSX-specific format:

node-id[/*user-id*[/*password*[/*account*]]]

where

node-id is the node name or address of the node to be accessed (see Section 2.1.1).

user-id is a string of up to 16 characters that is the user's log-in identification to be verified by the destination node. The log-in identification for RSX can be the account name or the account number. An account number can be specified in two ways: with brackets or without brackets. For example, the same account number could be expressed as [100,3] or 100,3.

password is a string of up to 8 characters to be verified against the destination node's system account file.

account is a string of up to 16 characters that supplies additional accounting information for the destination node. This field is ignored by RSX nodes.

If any string contains a blank character or tab, enclose the string with quotation marks ("*string*"). If you want to use a quotation mark within a quoted string, use double quotation marks (" ") to distinguish it from a string delimiter.

NOTE

If you do not want a password echoed to your terminal as you enter an NCP command, enter a carriage return after the keyword PASSWORD. NCP prompts for the password and turns off echoing until the next time it prompts.

2.1.1.2 Alias Node Names — One way to simplify the process of specifying access control is to set up alias node names that include access control information. Alias node names are pseudonames that you can use in place of a node ID whenever you identify a DECnet node in the network. Aliases can also be used by other DECnet programs and utilities (for example, NFT).

You can create, modify, display, and clear alias node names in the volatile data base and in the system image file by using the NCP/VNP SET ALIAS, SHOW ALIAS, and CLEAR ALIAS commands (see Chapter 6 for complete details). The following example shows how to set an alias node name:

```
NCP>SET ALIAS VAX DESTINATION CHI/[1,1]/DECNET TERMINAL TT11:
```

This command equates the alias name VAX with node CHI. Once this command has been executed, the alias name VAX can be used in NCP commands where you would normally have to specify CHI and its access control information. The above command specifies that the alias is valid on operations initiated from terminal TT11:. Therefore, all logical link connect requests from programs associated with terminal TT11: to node VAX are transmitted to node CHI. Similarly, an NCP>SHOW NODE VAX command will display data for node CHI.

Privileged users can specify aliases for a global scope — that is, for all terminals on the local node. Nonprivileged users can set aliases only for their own terminals (default). Any alias must be unique within its scope. Alias names are known only to your local node. DECnet-RSX software handles the translation from alias to node name.

DECnet-RSX automatically defines a blank alias node name for all network users. Connects to a blank node name default to the local node. You can change this association by setting a destination node for the blank alias — for example,

```
NCP>SET ALIAS "" DESTINATION CHI
```

The blank node name is useful for debugging network programs locally.

2.1.2 Specifying Access Control Verification

You can control inbound access to your node on two levels: the node level and the object level. You can specify that access control be verified for your node on all incoming connect requests to DECnet objects. In fact, node level access control verification is required on systems that support multiuser protection. To assure access control verification, you must set VERIFICATION parameters to ON at both the executor and the object level.

You can use CFE, NCP, and VNP commands to modify access control on both the node and the object level. If you set the VERIFICATION parameter to OFF in a SET/DEFINE EXECUTOR command, the executor overrides the verification options set for individual objects and allows access to all objects. When verification at the executor level is set to ON, verification states specified for individual objects are used (see the DEFINE/SET OBJECT command descriptions in Chapter 6).

CAUTION

> If access control verification is turned off, anyone can access your system.

2.1.3 Node Parameters

The permanent data base must contain certain information about the local node. Optionally, it can contain information for all nodes with which you wish to communicate. You can specify names, access control information, node type, and node counter information for any or all of the nodes. If a remote node can be down-line loaded, you can specify a number of default parameters to be used locally to perform a down-line load operation. Table 2-1 lists all node parameters by function and groups them according to the kind of node to which they apply.

You can modify and display node parameters by using the following commands (which are described with their parameters in Chapter 6):

CFE Commands	NCP/VNP Commands
DEFINE EXECUTOR	SET EXECUTOR
DEFINE NODE	SET NODE
—	CLEAR EXECUTOR
PURGE NODE	CLEAR NODE
LIST EXECUTOR	SHOW EXECUTOR
LIST NODE	SHOW NODE

2.1.3.1 Executor Node Identification String — During network generation, you can provide a string of information that is displayed whenever you display executor node information. To modify this string, use the IDENTIFICATION parameter with the DEFINE EXECUTOR command. If the string you specify contains space or tab characters, you must delimit the string with quotation marks. If you want to include a quoted string within a displayed string, use double quotations around the quoted string to distinguish it from the string's end delimiter. When the message is displayed, the system will discard one set of quotation marks.

Example:

```
CFE>DEFINE EXECUTOR IDENTIFICATION "DECnet RSX""BOS""V4.0"
```

Table 2-1: Node Parameters and Their Functions

Function	Executor Node	Remote Node
Node identification	ADDRESS NAME IDENTIFICATION	ADDRESS NAME
Local node state	STATE (ON/OFF/SHUT)	
Access control	VERIFICATION STATE (OFF/ON)	
Logical link control	MAXIMUM LINKS MAXIMUM NODE COUNTERS SEGMENT BUFFER SIZE (1)	
Routing control (2)	MAXIMUM ADDRESS MAXIMUM COST MAXIMUM HOPS MAXIMUM BROADCAST ROUTERS MAXIMUM BROADCAST NONROUTERS ROUTING TIMER BROADCAST ROUTING TIMER	
Routing initialization passwords (2)	TRANSMIT PASSWORD RECEIVE PASSWORD	
Loop node identification		CIRCUIT
Down-line loading (3)		DIAGNOSTIC FILE HARDWARE ADDRESS HOST LOAD FILE SECONDARY LOADER SERVICE CIRCUIT SERVICE DEVICE SERVICE NODE VERSION (PHASE III/PHASE IV) SERVICE PASSWORD TERTIARY LOADER
Up-line dumping (3)		DUMP ADDRESS DUMP COUNT DUMP FILE HARDWARE ADDRESS
Incoming PSI call control	SUBADDRESSES *range*	

1. See Chapter 5 for more information on specifying system buffers.
2. See Section 2.2 for a discussion of routing and routing parameters.
3. See Chapter 4 for information on down-line loading and up-line dumping.

2.1.3.2 Executor Node Subaddresses — During network generation, you can define a range of subaddresses for the executor node to use for DLM operations. The executor node accepts incoming DLM calls that have a subaddress that falls within the specified range.

Use the SUBADDRESSES parameter to modify the executor subaddresses in any of the three data bases. A subaddress can be a single number or a range of numbers in the range of 0 to 9999. When specifying a range, separate the two subaddresses that delimit the range with a hyphen, always listing the smaller number first.

Example:

```
NCP SET EXECUTOR SUBADDRESSES 42-50
```

This command specifies the executor node to handle all incoming PSI calls that specify a local DTE subaddress in the range of 42 to 50. All other calls will be handled by PSI.

2.1.4 Node Counters

DECnet software automatically collects certain statistics, called counters, for nodes in the network. Such information can include the number of connects sent and received, the number of messages sent and received, and the number of messages retransmitted.

If a logical link is established to a new node after all node counters have been used, the least recently used counter block is reassigned to the new node, and the counters for the old node are displayed in an event message.

For more information on using counters, see Section 2.7. For a complete listing of node counters, see Appendix E.

2.1.5 Ethernet Address of Node

Nodes on Ethernet lines are identified by unique Ethernet addresses. A message can be sent to one, several, or all nodes on an Ethernet line simultaneously, depending on the type of Ethernet address used. You do not normally have to specify the Ethernet address of an individual node in order to configure your network; the software at the node sets its own Ethernet address. You need to know the Ethernet address of a node for service functions (such as down-line load or circuit loopback test) but not for normal network operations.

2.1.5.1 Format of Ethernet Addresses — Ethernet addresses are represented as six pairs of hexadecimal digits separated by hyphens (for example, AA-01-23-45-67-FF).

Xerox Corporation assigns a block of addresses to a producer of Ethernet interfaces upon application. Thus every manufacturer has a unique set of addresses to use. Normally, one address out of the assigned block of physical addresses is permanently associated with each interface (usually in a read-only memory). This address is known as the **Ethernet hardware address** of the interface.

NOTE

> You can use the NCP SHOW LINE *line-id* CHARACTERISTICS command to display the hardware address. See the example in Chapter 3.

Digital's interface to Ethernet (the DEUNA or DEQNA controller at the node) has the ability to set a different logical address to be used by the interface: this address is known as the **Ethernet physical address**. When a node on the Ethernet initially starts up, the physical address is the same as the Ethernet hardware address. Then when DECnet turns on an UNA or QNA device, the DEUNA or DEQNA controller constructs a physical address by appending the local node's 4-digit node address to a constant 8-digit number derived from the block addresses assigned to Digital (AA-00-04-00).

Once the Ethernet physical address has been set to its new value, it is reset to its original hardware address value only when a reset is issued to the DEUNA or DEQNA (for example, when the machine power is shut off).

2.1.5.2 Ethernet Multicast Address Types — Ethernet physical addresses (described above) and Ethernet multicast addresses (described below) are distinguished by the the value of the leading low-order bit of the first byte of the address:

- **Physical address.** The unique address of a single node on any Ethernet (low-order bit = 0).
- **Multicast address.** A multidestination address of one or more nodes on a given Ethernet (low-order bit = 1).

There are two types of multicast address:

- A **multicast group address** is an address that is assigned to any number of node groups so that they are all able to receive the same message in a single transmission by a sending node.
- A **broadcast address** is a single multicast address (specifically, FF-FF-FF-FF-FF-FF) to which a message can be sent if it must be received by all nodes on a given Ethernet. (Use a broadcast address only for messages to be acted upon by all nodes on the Ethernet, since all nodes must process them.)

2.1.5.3 Ethernet Physical and Multicast Address Values — Digital physical addresses are in the range AA-00-00-00-00-00 through AA-00-04-FF-FF-FF. Multicast addresses assigned for use in cross-company communications are:

Value	Meaning
FF-FF-FF-FF-FF-FF	Broadcast
CF-00-00-00-00-00	Loopback assistance

Digital multicast addresses assigned to be received by other Digital nodes on the same Ethernet are:

Value	Meaning
AB-00-00-01-00-00	Dump load assistance
AB-00-00-02-00-00	Remote console
AB-00-00-03-00-00	All Phase IV routers
AB-00-00-04-00-00	All Phase IV end nodes
AB-00-00-05-00-00 through AB-00-04-FF-FF-FF	Reserved for future use

DECnet always sets up the DEUNA or DEQNA at each node to receive messages sent to any address in the above list of Digital multicast addresses.

2.2 Routing

Routing is the network function that determines the path or route along which data (called packets in this context) travels from its source to its destination. Routing is performed transparently by the Transport layer of DECnet software. As system manager, however, you must be concerned with the configuration of the network into routing and nonrouting nodes and with a set of parameters that provides a degree of indirect control over network routing. This section explains the different types of routing and nonrouting nodes and then describes the routing parameters you can control. The *Introduction to DECnet* and the *Overview of DECnet-RSX* provide more detailed information about routing concepts and mechanisms.

2.2.1 Types of Nodes

A Phase IV DECnet-RSX node can be either a routing node or an end node. Either type of node can be connected to a DDCMP, PCL, Ethernet, or DLM circuit. All Phase IV nodes can communicate with one another and with Phase III nodes. Phase IV nodes, however, cannot communicate with Phase II nodes.

Routing nodes have a route-through capability and are able to support multiple circuits. They maintain routing data bases that allow them to route their own transmit packets as well as packets received from other nodes. On an Ethernet, a single routing node called the designated router (see Section 2.5.9) performs routing functions for all of the nodes on the Ethernet.

End nodes support only one circuit and do not maintain a routing data base. An end node can establish logical links to any node in the network, but it can send packets to and receive packets from the adjacent node only. Because all nodes on an Ethernet are considered to be adjacent to each other, an Ethernet end node can communicate directly with more than one node. End nodes connected via a DDCMP, PCL, or DLM circuit, however, are adjacent to a single node only.

2.2.2 Routing Parameters

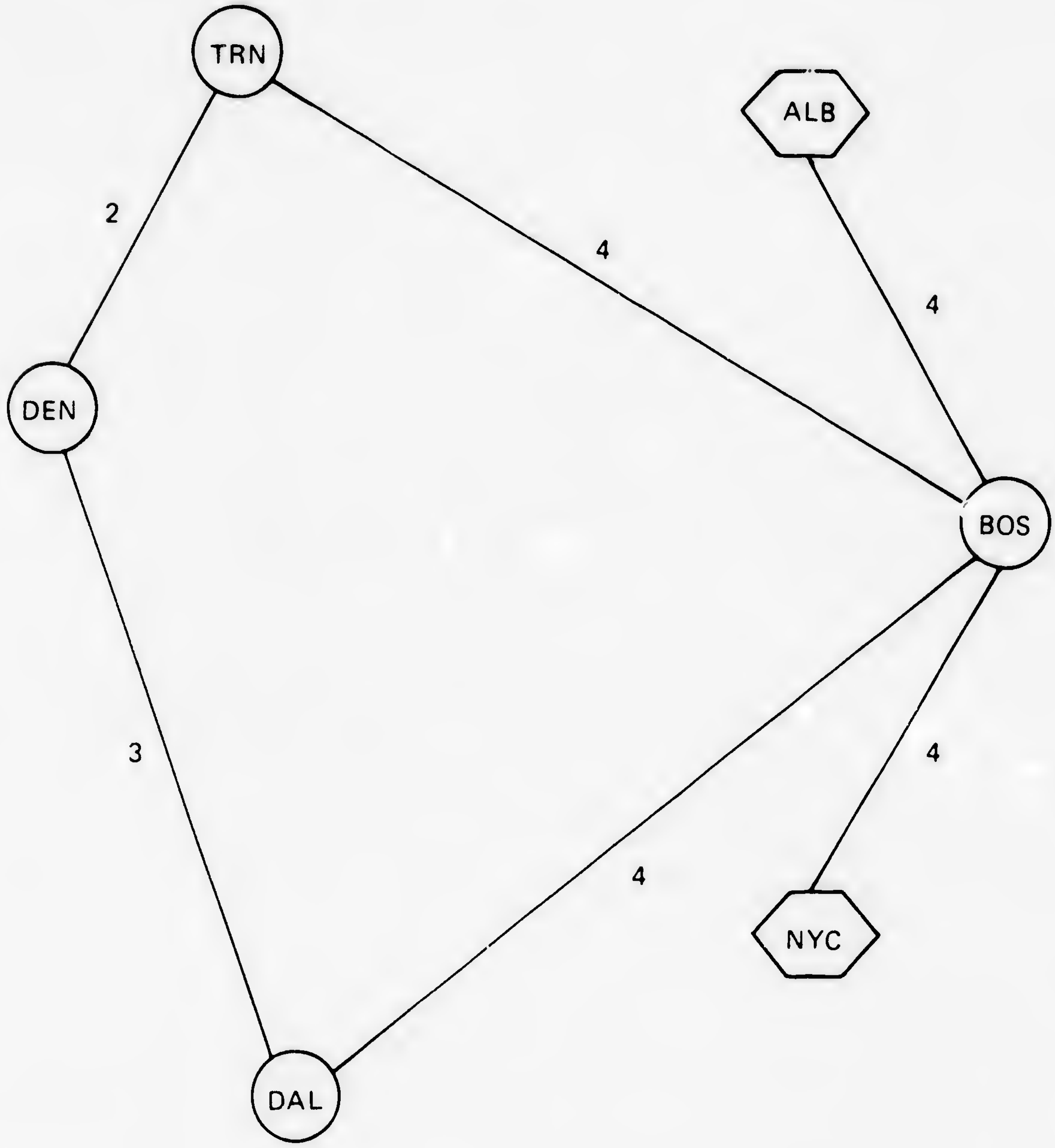
Some parameters that affect routing are defined for the executor node component, while others are defined for the circuit component. Table 2-2 lists the parameters by component and specifies the utilities that can modify them. Parameters that apply to Ethernet nodes only are listed separately. The network generation procedure sets default values for all of these parameters.

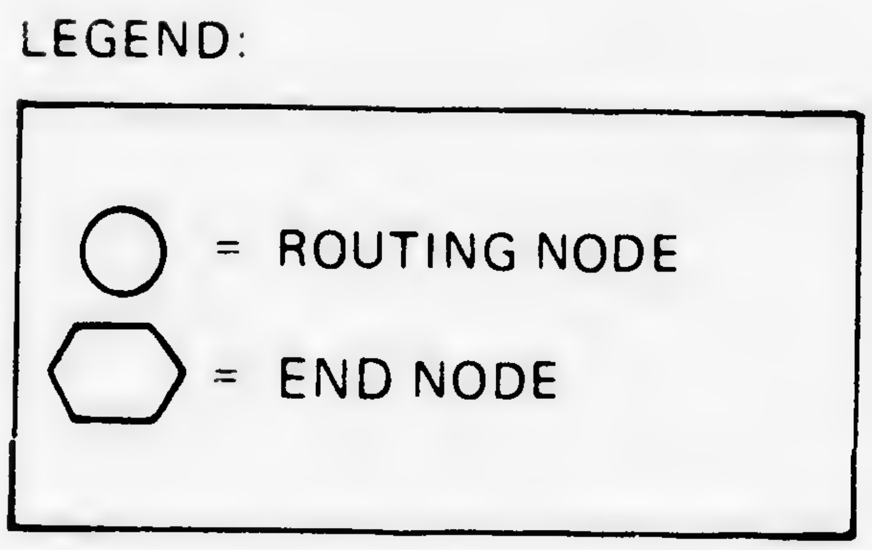
Table 2-2: Routing Parameters

Component	Parameter	CFE	NCP/VNP
General routing parameters:			
EXECUTOR	MAXIMUM ADDRESS	X	
	MAXIMUM COST	X	
	MAXIMUM HOPS	X	
	ROUTING TIMER	X	X
CIRCUIT	COST	X	X
	HELLO TIMER	X	X
Ethernet-only parameters:			
EXECUTOR	BROADCAST ROUTING TIMER	X	
	MAXIMUM BROADCAST NONROUTERS	X	
CIRCUIT	MAXIMUM BROADCAST ROUTERS	X	
	ROUTER PRIORITY (described in Section 2.5.9)	X	

2.2.2.1 Maximum Node Address — During network generation, you specify the highest node address to be allowed in the network. This controls the size of routing configuration messages exchanged between nodes and determines the size of the internal routing data base constructed by the network software. You can modify this value in the permanent data base using the MAXIMUM ADDRESS parameter. To obtain the most efficient routing operation, sequentially number your nodes from 1 to *n*, where *n* is the highest node address.

2.2.2.2 HELLO TIMER Parameter — The DECnet Transport layer software sends hello messages over DECnet circuits at regular intervals and waits for a response to determine if the circuit is still operative. If the circuit is inactive, the Transport layer attempts to reinitialize it. You can use the HELLO TIMER parameter to modify the frequency with which messages are transmitted.





PATH	LINE COST	PATH COST	HOPS
#1	TRN→DEN→DAL 2 3	5	2
#2	TRN→BOS→DAL 4 4	8	2

Figure 2-1: Circuit and Path Cost Relationship

2.2.2.3 Circuit Cost — During network generation, a cost is automatically assigned to each circuit connecting the local node with an adjacent node. DECnet software uses circuit cost values to determine the path over which data is transmitted. You can use the COST parameter to change the cost of a circuit. Altering circuit costs can change packet-routing paths and thereby affect the use and availability of network circuits and resources.

Use the following guidelines when first assigning circuit costs; you can adjust these values later if doing so will improve performance.

Ethernet circuits	3
Line speeds of 9600 or less	5
Line speeds over 9600	7
DLM circuits	15

Figure 2-1 illustrates the relationship between circuit and path costs.

2.2.2.4 MAXIMUM COST and MAXIMUM HOPS Parameters — You can alter the values set for the MAXIMUM COST and MAXIMUM HOPS parameters, which determine the reachability of nodes in the network.

The MAXIMUM COST parameter specifies the maximum total path cost allowed from the executor node to any other network node. A remote node is unreachable if the cost to get to the remote node exceeds the value set for this parameter. Use as small a number as possible in the range of 1 to 1022.

The MAXIMUM HOPS parameter specifies the maximum number of hops allowed from the executor to any other node in the network. The largest value for this parameter in a DECnet network defines the DECnet network diameter. A remote node is unreachable if the number of hops required to reach it exceeds the value set for this parameter. Use as small a number as possible in the range of 1 to 30.

2.2.2.5 Routing Timers — DECnet-RSX provides a timing mechanism that causes circuit, line, and node configuration changes and routing configuration messages to be transmitted to all adjacent nodes from the local node. Routing configuration messages provide dynamic network configuration information that can affect data packet routing. For example, if a network user changes the state of a circuit and thereby renders a remote node unreachable, this change is reflected automatically in the routing configuration data transmitted to adjacent nodes.

On non-Ethernet nodes, you use the ROUTING TIMER parameter to set a timer whose expiration forces a routing update regardless of whether the network configuration has altered.

For a node on an Ethernet circuit, the timing mechanism is called a broadcast routing timer. When the timer expires, the local node sends a multicast routing configuration message to all nodes on the Ethernet (since all nodes are considered to be adjacent). You can use the BROADCAST ROUTING TIMER parameter to specify the frequency at which these messages are transmitted. This timer is usually set to a much lower value (approximately 30 to 40 seconds) than the routing timer for a node on a non-Ethernet circuit (which is set for every few minutes). Ethernet routing messages are sent more often, so that full routing messages can be exchanged in case of datagram loss.

2.3 Objects

An object is a DECnet user task or module that can communicate with another such task or module over a logical link. All objects are identified by an object type code of either 0 or an integer in the range of 1 to 255, depending on the kind of object.

- **Named objects** are installed tasks identified by any valid user-defined RSX task name in a logical link connect request. All named objects have an object type code of 0.
- **Numbered objects** are installed tasks or modules (such as FAL and NICE) that provide network services. Each numbered object is identified by an object type code in the range of 1 to 255. Each type code always represents the same function within a network, even if the task that actually performs the function has a different name from node to node. Numbers in the range of 1 to 127 are reserved for Digital's use; numbers from 128 through 255 are available for customers. Appendix B lists all DECnet-RSX object type codes.

At network generation, the system automatically defines objects for all selected tasks. Later, you can use the DEFINE/SET commands to create or modify a task or object, using your own value or an object type code from Appendix B to specify the object. Parameters that you can specify for each type of object are summarized in Table 2-3.

Table 2-3: Object Parameters

Parameter	Named Objects	Numbered Objects
COPIES		X
NAME		X
USER	X	X
VERIFICATION	X	X

Since all named objects have the same type code (0), any changes that you make to object type 0 apply to all named objects on your node. For programming information on named objects and their use in task-to-task communication, refer to the *DECnet-RSX Programmer's Reference Manual* and the *RSX-11 PSI User's Guide*.

In this manual, you will find descriptions of these parameters under the following commands in Chapter 6:

CFE Commands	NCP/VNP Commands
DEFINE OBJECT	SET OBJECT
PURGE OBJECT	CLEAR OBJECT
LIST OBJECT	SHOW OBJECT

2.3.1 Single Copy and Multicopy Objects

Use the COPIES parameter to specify the number of copies of a task that can be run at the local node at once. Specify SINGLE for one copy or specify a number in the range of 2 to 64 for a multicopy object. All numbered object types can have multiple copies.

When you specify a number, each incoming request is sent to a new copy of the task. Thus, each copy of the task handles one incoming connection. This is referred to as single-threaded operation.

When you specify SINGLE, the task determines the number of simultaneous connections it can have. This is referred to as multithreaded operation.

2.3.2 Object Names

When you install a task as a numbered object, you can use the NAME parameter to specify any task name that is valid for RSX. When a numbered object has multiple copies, you must install it with a name in the following format: *xxx*$$$, where *xxx* is any valid 3-character RSX task name. If the task name is not installed as such, a logical link connection cannot be established.

When the network software establishes the logical link, it replaces the three dollar signs with a specific number. You can see the final task names in the NTD display of active tasks (see Chapter 11).

2.3.3 Object UICs

Once activated, objects run under a particular UIC. You can use the USER parameter to specify whether the object runs under the log-in UIC or under the default UIC under which it was built or installed. The log-in UIC is the user ID that was specified in the access control information provided at the time of the connection. If a log-in UIC is not provided, the object runs under the default UIC.

The object's access control verification option must be set to ON or INSPECT to run under log-in UICs. If the object's access control verification option is not on, the object runs under the default UIC. See Section 2.1.2 for a discussion of object verification options and access control.

2.4 Lines

DECnet-RSX supports four types of lines: DDCMP, PCL, Ethernet, and PSI. A DDCMP line provides the physical point-to-point or multipoint connection between two or more nodes; PCL and Ethernet lines are multiaccess connections between two or more nodes; and a PSI line is the physical link between a user's data terminal equipment (DTE) and a packet-switching network (PSN).

Just as you must establish node parameters, you must also establish parameters for all physical lines connected to the local node. Thus you must identify each line by name and specify information that directly affects the line's operation.

2.4.1 Lines and Line Devices

Table 2-4 lists the devices that operate on the four types of lines supported by DECnet-RSX (DDCMP, PCL, Ethernet, and PSI). Each line type is described in the following sections. You should be familiar with the entire range of devices and their impact on network management. For additional information on these line devices, refer to the *Terminals and Communications Handbook*.

2.4.1.1 DDCMP Line Devices — Four of the DDCMP line devices listed in Table 2-4 have the DDCMP microcode contained in the hardware device: DMC-11, DMR-11, DMP-11, and DMV-11. The other DDCMP devices in Table 2-4 have a software DDCMP process.

The DMC-11 and the DMR-11 (which operate identically) provide a point-to-point connection between two nodes. Circuits, the actual communications path, operate over the line.

The DMP-11 and the DMV-11 can be point-to-point, multipoint control, or multipoint tributary line devices. Therefore, they can provide a multipoint connection between more than two nodes. It is possible to connect two multipoint lines to the same node. The node can then serve as the control station for one multipoint line and as a tributary for another multipoint line.

2.4.1.2 PCL Line Devices — The PCL-11B provides point-to-point connections by means of a parallel time division multiplexed (TDM) bus. PCL-11B hardware allocates a fraction of available bus bandwidth to each station.

2.4.1.3 Ethernet Line Devices — The Ethernet line device is either the UNA or the QNA, which provides a physical connection between two or more nodes on the same Ethernet cable. The Ethernet line protocol supports up to 1023 nodes on the same line. The Ethernet circuit operates over the Ethernet line.

A node is connected to the Ethernet line by a DEUNA or DEQNA communications controller, a transceiver, and a transceiver cable. A specific node on the Ethernet is identified by the Ethernet hardware address of its line device;

Table 2-4: Digital Communications Devices Supported by DECnet-RSX

Device	Line Type	Multiline	Multipoint
DL	DDCMP	No	Yes
DLV	DDCMP	No	Yes
DMC	DDCMP	No	No
DMP	DDCMP	No	Yes
DMR	DDCMP	No	No
DMV	DDCMP	No	Yes
DPV	DDCMP	No	Yes
DU	DDCMP	No	Yes
DUP	DDCMP	No	Yes
DUV	DDCMP	No	Yes
DV	DDCMP	Yes	Yes
DZ	DDCMP	Yes	Yes
DZV	DDCMP	No	Yes
KDP	DDCMP	Yes	Yes
KDZ	DDCMP	Yes	Yes
KMX	PSI	Yes	N/A
PCL	PCL	No	Yes
SDP	PSI	No	N/A
SDV	PSI	No	N/A
UNA	Ethernet	N/A	N/A
QNA	Ethernet	N/A	N/A

this hardware address is stored in read-only memory in the DEUNA or DEQNA. When DECnet starts an Ethernet line, it causes the DEUNA or DEQNA connected to the line to construct an Ethernet physical address for the node (see Section 2.1.5.1). The DEUNA or DEQNA resets the physical address to the original hardware address if the device is reset (for example, if the machine power is shut off).

2.4.1.4 PSI Line Devices — RSX-11 PSI supports three line devices: the DUP11-DA, the KMS11-BD, and the KMS11-PX. The DUP11-DA is a low speed synchronous interface. The combination of the KMS11-BD controller hardware and the X.25 level 2 microcode provides a medium speed synchronous interface called the KMX. The KMS11-BD supports eight lines, only two of which can be active simultaneously. Similarly, the combination of the KMS11-PX controller hardware and the X.25 level 2 microcode provides a medium speed synchronous interface called the KMX.

2.4.2 Line Identification

You can use network management commands to modify parameters for an individual line or for all known lines. Use the KNOWN LINES keyword to specify all lines identified to the system during network generation. To specify an individual line, use a line ID of the form

dev-c[-u]

where

dev	is a device name from Table 2-4.
c	is a decimal number (0 or a positive integer) designating the device's hardware controller.
u	is a decimal unit number (0 or a positive integer) included if the device is a multiple line controller.

You can use a wildcard character (*) in place of the controller or unit number to specify all lines in that category. You cannot specify a particular unit if you specify a wildcard controller.

Tributaries for a multipoint line device are defined at the circuit level (see Section 2.5.6.1).

NOTE

> To create a new line ID in the configuration file, you must perform another network generation.

2.4.3 Line Parameters

The permanent data base should contain line parameters for all physical lines connected to the local node or DTE. These parameters supply information used to control various aspects of a line's operation. You can modify and display line parameters by using the following commands (which are described with their parameters in Chapter 6):

CFE Commands	NCP/VNP Commands
DEFINE LINE	SET LINE
PURGE LINE	CLEAR LINE
LIST LINE	SHOW LINE

NCP/VNP line parameters fall into two categories: loaded options and loading options. Loaded options are valid only for lines that have already been loaded into the volatile data base. Loading options apply to lines that are currently being loaded.

Table 2-5 categorizes all line parameters according to line type and parameter function. Additional notes on some parameters are included in the following sections, but you should see Chapter 6 for complete details.

Table 2-5: Line Types and Parameter Functions

Line Types and Parameter Functions	Parameters
All lines	
Specifies device hardware characteristics	CONTROLLER CSR UNIT CSR VECTOR PRIORITY SPEED
Sets line's operational state (valid for all lines on CFE commands, for PSI only on NCP/VNP)	STATE (CLEARED/OFF/ON)
Sets hardware transmission mode	DUPLEX (FULL/HALF)
Establishes line level loopback control for device (applies to DMC, DMR, DMP, DMV, UNA, QNA lines only)	CONTROLLER (LOOPBACK/NORMAL)
Specifies where to load software	LOCATION (FIRSTFIT/TOPDOWN)
Multipoint DDCMP lines	
Establishes DDCMP line-polling protocol (see parameter descriptions in Section 2.5.8)	DEAD TIMER DELAY TIMER MULTIPOINT DEAD
PSI lines	
Sets counter timer for event logging	COUNTER TIMER
Establishes frame control	HOLDBACK TIMER MAXIMUM DATA MAXIMUM WINDOW
Controls retransmission of frames	RETRANSMIT TIMER MAXIMUM RETRANSMITS
Changes line's owner	OWNER (DLX/PLI)

2.4.3.1 Hardware Device Parameters — During network generation, you specified a number of parameters that directly affect the operation of the hardware device for a line. You can modify this information as necessary in any of the three data bases using network management commands. If you change the controller CSR address on a line, be sure to make the same change for any other lines sharing that controller.

On KMC-11 devices (KDP and KDZ), use the UNIT CSR parameter to modify the address of the first CSR. You must also specify the CSR of the secondary device (DUP or DZ) that the KMC is controlling. In addition, you can use the UNIT CSR parameter to modify the CSR for the KMX modem controller.

Use the VECTOR parameter to modify the address of the vector that contains the address of the interrupt service routines of a line.

2.4.3.2 Line States and Loading — DECnet users indirectly set line states by setting the states of DECnet circuits (see the DEFINE/SET commands in Chapter 6). PSI lines can be set to the same states permitted for DECnet circuits, except that PSI line states are specified in line commands (see SET LINE in Chapter 6).

You can use the SHOW LINE command (see Chapters 3 and 6) to display the state of any or all lines. In some cases, a substate is also displayed. Circuit/line states and substates are described in Section 2.5.7.1 and are summarized in Table 2-7.

Line loading is directly related to the line states specified either in the permanent data base or during network generation. Lines with a CLEARED state are not loaded when you load the DECnet-RSX system. Lines with an OFF state are loaded, but must be turned on in the volatile data base (PSI lines are turned on at the line level, and non-PSI lines are turned on at the circuit level — see Section 2.5.7.1).

2.4.3.3 PSI Parameters — The following considerations should be reviewed when modifying the related parameters.

Frame control. In the permanent data base, the maximum data and window size parameters were set to defaults for the PSN to which the line's DTE is connected. Consult the *RSX-11 PSI Network-specific Information* manual for details of these values before you modify them.

Frame retransmission. When you specify a value for the retransmit timer, be sure to account for the speed of the line and the block size.

2.4.4 Line Counters

DECnet software automatically maintains statistics, called counters, for most lines in the network. Line counters for DDCMP lines include the number of bytes and data blocks sent and received, local and remote process errors, and the amount of time since the counters were last zeroed. DECnet-RSX maintains these counters for all DDCMP lines except DMC/DMR and DMP/DMV point-to-point lines.

Line counters for Ethernet lines include the number of bytes, multicast bytes, data blocks, and multicast blocks sent and received; the number of blocks deferred or sent after collision; and the number of send failures and discarded frames.

NOTE

> The UNA returns line counter information only if a circuit is active on the line.

RSX-11 PSI line counters include such information as bytes and data blocks sent and received, inbound and outbound data errors, and local and remote reply timeouts, buffer errors, and process errors. These counters, together with component characteristics, are useful in monitoring the activity of PSI lines.

For more information on using counters, see Section 2.7. For a complete listing of line counters, see Appendix E.

2.5 Circuits

Circuits are high level communications paths between nodes or DTEs (data terminal equipment supported by PSI). Circuits are the logical connections between two nodes (or all nodes — on an Ethernet cable) that operate over the physical medium of lines.

DECnet-RSX supports five types of circuits: DDCMP, PCL, Ethernet, DLM, and PSI. Each is briefly described below. Circuit ID formats are outlined later in this chapter. For more information on circuits, refer to the *Introduction to DECnet*.

2.5.1 DDCMP Circuits

DDCMP circuits provide the logical connection between two adjacent nodes. They are classified according to the type of line over which they operate:

- A **point-to-point circuit** connects two nodes over a corresponding point-to-point line.
- A **multipoint circuit** operates over a multipoint line, which serves more than two nodes. On a multipoint line, the **multipoint control circuit** is affiliated with the line's control station. In addition, there is one **multipoint tributary circuit** for every tributary on the line.

You can specify multiple circuits from the control (master) end of a control line, with each circuit having a unique physical tributary address. From the tributary (slave) end, you have only one multipoint tributary circuit per line.

2.5.2 PCL Circuits

PCL circuits provide point-to-point connections between up to 16 nodes on a line by means of a parallel time division multiplexed (TDM) bus. The PCL-11B hardware allocates a fraction of available bus bandwidth to each station.

Each PCL-11B node is configured as a control station. From the perspective of any PCL-11B node on the bus, other PCL-11B nodes appear to be tributaries. However, the concepts of active, dying, and dead tributaries do not apply. Each station is assigned a unique tributary address, which is established when the hardware is installed.

2.5.3 Ethernet Circuits

Ethernet circuits provide for multiaccess connection among all of the nodes on the same Ethernet cable. An Ethernet circuit differs from other DECnet circuits in that you connect not to a single node, but to many. Each node on a single Ethernet circuit is considered to be adjacent to every other node on the circuit and equally accessible. For detailed information on Ethernet, refer to the *Networks: Ethernet Products and Services Catalog*.

2.5.4 DLM Circuits

A data-link-mapping (DLM) circuit is a DECnet circuit that uses PSI facilities to connect DECnet nodes. DLM supports two types of PSI circuit:

- **Permanent virtual circuit (PVC).** Provides a permanent path between the local DTE and the remote DTE (analogous to a leased line).
- **Switched virtual circuit (SVC).** Provides a temporary path between the local DTE and the remote DTE (analogous to a dial-up line). An SVC is set up using network management commands only when there is data to transmit and is cleared when the transfer is complete. The SVC can be designated exclusively for incoming or outgoing calls (see DEFINE CIRCUIT USAGE).

2.5.5 PSI Circuits

A PSI circuit is a virtual circuit connecting local data terminal equipment (DTE) with a remote DTE. PSI circuits operate differently from DECnet circuits. All PSI circuits pass through the X.25 protocol module (see Section 2.9.1), which multiplexes circuits to lines that it owns. As such, there is no direct relationship between the name of a PSI circuit and a PSI line, other than the arbitrary one that designates whether the circuit is used for DECnet or for PSI network operations.

PSI supports the use of both PVCs and SVCs (see definitions in Section 2.5.4). Native PSI SVCs are set up with parameters taken from the X.25 protocol module component when calls are requested on these circuits. Refer to Chapter 6 for details on specifying parameters for PVCs in CFE, NCP, and VNP commands.

PSI circuits are used in two ways:

- For user application programs (PSI native circuits).
- For the mapping of data link information from the DECnet Transport layer via the X.25 protocol module of the local DECnet/PSI node to the X.25 protocol module of a remote DECnet/PSI node. DLM (see Section 2.5.4) enables the use of the X.25 protocol for DECnet node-to-node communication.

2.5.6 Circuit Identification

You can use network management commands to modify parameters for an individual circuit or for all known circuits. Use the KNOWN CIRCUITS keyword to specify all circuits identified to the system during network generation (including circuits that are active and circuits whose state is OFF). To specify an individual circuit, use the CIRCUIT keyword and a circuit ID. DECnet and PSI circuits have different formats, as described in the following sections.

NOTE

> You can modify circuit parameters only for DECnet circuits defined in the permanent data base during network generation. If you want to create another circuit ID in the permanent data base, you have to do another network generation.

2.5.6.1 DECnet Circuit Identification (DDCMP, PCL, Ethernet circuits) — To specify an individual DECnet circuit, use a circuit ID of the form

dev-c[*-u*][*.t*]

where

dev is a device name from Table 2-4 (for example, DMC, PCL, UNA).

c is a decimal number (0 or a positive integer) designating the device's hardware controller.

u is a decimal unit/circuit number (0 or a positive integer) included if the device is a multiple line controller.

t is a decimal number identifying a tributary circuit on a multipoint line.

You can use a wildcard character (*) in place of the controller, unit, or tributary number to specify all known circuits in that category. You can use multiple wildcards in a circuit ID, as long as no numbers follow them. Some examples of DECnet circuit IDs follow:

Circuit ID	Meaning
UNA-1	UNA, controller 1
DZ-0-4.2	DZ, controller 0, unit 4, tributary 2
DZ-0-4.*	DZ, all known tributaries on unit 4 of controller 0
DL-1.3	DL, controller 1, tributary 3
PCL-0.*	PCL, all known tributaries on controller 0

Table 2–6: Circuit Types and Parameter Functions

Circuit Types and Parameter Functions	Parameters
All circuits (except PSI)	
Sets circuit's operational state	STATE (CLEARED/OFF/ON/SERVICE)
Specifies circuit owner	OWNER (DLX/XPT)
Specifies circuit cost for routing	COST *
Sets frequency of routing on circuit	HELLO TIMER *
Specifies whether down-line loading and loopback testing are enabled	SERVICE (DISABLED/ENABLED)
Multipoint DDCMP circuits	
Sets tributary address for circuit	TRIBUTARY
Specifies tributary polling rate	MULTIPOINT ACTIVE
PCL circuits	
Sets tributary address for circuit	TRIBUTARY
Ethernet circuits	
Specifies router parameters for circuit	MAXIMUM BROADCAST ROUTERS ROUTER PRIORITY
DLM circuits (PVCs or SVCs)	
Controls data packets for PVCs and SVCs	MAXIMUM DATA MAXIMUM WINDOW
Specifies channel and DTE for a PVC	CHANNEL DTE
Assigns remote DTE address for an SVC	NUMBER
Controls retransmission when establishing an SVC	MAXIMUM RECALLS RECALL TIMER
Specifies availability of an SVC for incoming or outgoing calls	USAGE (INCOMING/OUTGOING)
PSI native PVCs	
Specifies channel and DTE for a PVC	CHANNEL DTE
Sets counter timer for event logging	COUNTER TIMER
Controls data packets for PVCs	MAXIMUM DATA MAXIMUM WINDOW

* See parameter descriptions in Section 2.2.2.

2.5.6.2 DLM Circuit Identification — Circuit IDs for DLM PVCs and SVCs have the form

DLM-*x.y*

where

x identifies a group code analogous to the controller number on a device. You can use this to group DLM circuits according to type (for example, all PVCs could have the same group code).

y identifies the circuit number analogous to the logical number associated with tributaries for a device.

2.5.6.3 PSI Circuit Identification — Circuit IDs for PSI PVCs consist of a string of 1 to 6 alphanumeric characters. You cannot identify SVCs for PSI circuits.

2.5.7 Circuit Parameters

The permanent data base contains circuit parameters for all circuits connected to the local node or DTE. You can modify and display circuit parameters by using the following commands (which are described with their parameters in Chapter 6):

CFE Commands	NCP/VNP Commands
DEFINE CIRCUIT	SET CIRCUIT
PURGE CIRCUIT	CLEAR CIRCUIT
LIST CIRCUIT	SHOW CIRCUIT

Table 2-6 categorizes all circuit parameters according to circuit type and parameter function. Additional notes on some parameters are included in the following sections, but you should see Chapter 6 for complete details.

2.5.7.1 Circuit States and Loading — You control the operational state of all circuits at the local node. This allows you to control circuit traffic and to perform service functions on DDCMP circuits. The state of a circuit can affect the reachability of an adjacent node. Use the STATE parameter to set one or all known circuits to one of the following states:

CLEARED In the permanent data base, this state means that the circuit is not loaded when the network is loaded and is unavailable for use, even though it is defined in the permanent data base.

In the volatile data base or the system image file, this state means that the associated line has not been loaded and is unavailable for use.

OFF In the permanent data base, this state means that the circuit is loaded when the network is loaded, but is not being used.

In the volatile data base or the system image file, this state means that the circuit is turned off.

ON	In the permanent data base, this state means that the circuit is loaded and available for use when the network is loaded. In the volatile data base or the system image file, this state means that the circuit is available for use.
SERVICE	This state exists only in the volatile data base and the system image file and is valid for DECnet nodes only. It means that the circuit is reserved for service functions such as up-line dumping, down-line system loading, or circuit level loopback testing.

Table 2-7: Circuit/Line States and Substates

State	Substate	Meaning
CLEARED	none	The line has not been loaded.
OFF	none	The circuit/line is not being used.
ON	-RUNNING (not displayed)	The circuit/line is in normal use.
	-STARTING	The circuit is in the (Transport) initialization cycle.
	-AUTOSERVICE	The circuit is reserved for automatic service use.
	-LOADING	The circuit is in use for loading.
	-AUTOLOADING	The circuit is in use for automatic loading.
	-DUMPING	The circuit is in use for dumping.
	-AUTODUMPING	The circuit is in use for automatic dumping.
	-TRIGGERING	The circuit is in use for triggering.
	-AUTOTRIGGERING	The circuit is in use for automatic triggering.
	-LOOPING	The circuit/line is in use for active loopback testing.
	-REFLECTING	The circuit/line is in use for passive loopback testing.
	-FAILED	The DLM circuit has unsuccessfully recalled an SVC the specified maximum number of times.
SERVICE	-IDLE (not displayed)	The circuit is reserved for an active service function.
	-LOADING	The circuit is in use for loading.
	-DUMPING	The circuit is in use for dumping.
	-TRIGGERING	The circuit is in use for triggering.
	-LOOPING	The circuit/line is in use for active loopback testing.
	-REFLECTING	The circuit/line is in use for passive loopback testing.

Circuit substates. When you use the NCP SHOW CIRCUIT or SHOW LINE commands (see Chapter 3), the system displays the state to which you have most recently set the circuit(s). In some cases, it also displays one of the substates listed in Table 2-7.

DECnet software can automatically change the state of a circuit for certain functions. For example, if you or a remote user initiate a loopback test over a circuit, DECnet network management software automatically changes the state of that circuit to the appropriate internal state (or substate).

Several circuit substates have an AUTO- prefix. These substates can occur when you have an adjacent node that is about to be or is in the process of being automatically down-line loaded or triggered. For example, if a circuit is on and the local node senses a request for a down-line load on that circuit, the network management software on the local node automatically sets the circuit to the ON-AUTOSERVICE state.

Table 2-7 lists all circuit states and substates and their meanings. These states and substates also apply to PSI lines. Refer to the *DNA Network Management Functional Specification* for further information about line states, substates, and their transitions.

2.5.7.2 Circuit Ownership — On each network node, there are processes that can own the different circuits and control traffic over them. Use the OWNER parameter in the SET CIRCUIT command to set the circuit owner to match the owner for the associated DECnet line. There are two possible owners for DDCMP circuits:

DLX The Direct Line Access Controller. When DLX owns a circuit, you can use the circuit as an I/O device (that is, you can direct circuit traffic at the Data Link level). The *DECnet-RSX Programmer's Reference Manual* describes the DLX user interface; the Data Link layer of the Digital Network Architecture is described in the *Introduction to DECnet*.

XPT The Transport layer. When XPT owns a circuit, a network user can perform normal logical link operations over it. The *Introduction to DECnet* describes the Transport layer of the Digital Network Architecture.

DECnet software can temporarily override XPT as the owner of a DDCMP circuit and reserve the circuit for the following service functions:

- Down-line system loading to a remote node
- Up-line dumping from a remote node
- Triggering a remote node
- Loopback testing for a circuit at the physical link level

In these situations, the circuit owner remains XPT, but XPT cannot use the circuit until the service function has finished using the circuit. DDCMP circuits can be reserved for the service functions listed above by setting the circuit to the SERVICE state in the volatile data base (see Section 2.5.7.1).

By default, the owner of DLM circuits is XPT, and the owner of PSI circuits is the NW process. Since Ethernet circuits allow access by multiple users, they do not have exclusive owners.

2.5.8 DDCMP Multipoint Circuit Parameters

DECnet-RSX supports DDCMP multipoint circuits for the following devices: DL-11, DLV-11, DMP-11, DMV-11, DU-11, DUP-11, DPV-11, DUV-11, DV-11, DZ-11, KDP-11, and KDZ-11. The following sections describe multipoint operation, including the parameters you can modify to control device operation. Multipoint characteristics are defined at both the line and the circuit level. Applicable parameters are:

For lines:	DEAD TIMER (for DMP and DMV devices only) DELAY TIMER (for DMP and DMV devices only) MULTIPOINT DEAD
For circuits:	MULTIPOINT ACTIVE TRIBUTARY

Circuit parameters apply to a specific tributary. Line parameters apply to all tributaries on the specified line(s).

2.5.8.1 Multipoint Operation — A multipoint circuit has one control station and multiple tributaries. The control station continually polls existing (that is, on-line) tributaries for data awaiting transmission. A tributary is always given the opportunity to respond when it is polled. If the tributary is active, it either transmits data or returns an ACK (positive acknowledgement) if it has no data to transmit. Whenever the control station has data for a tributary, it transmits the data immediately.

An active tributary is one with the circuit turned on. If the control station does not get a response for two consecutive polls, it considers the tributary to be dying and polls less frequently. If the tributary does not then respond to the next six polls, the control station considers it to be dead and polls it even less frequently. If a dead tributary begins to respond, it is reclassified as active and is polled at the original active polling rate.

DECnet software uses polling ratios to control polling rates for active and dead tributaries. The polling ratios have system defaults, but you can use the commands SET/DEFINE CIRCUIT MULTIPOINT ACTIVE and SET/DEFINE LINE MULTIPOINT DEAD (except for DMPs and DMVs) to modify them (see Chapter 6 for details). For DMPs and DMVs, you must use the DEAD TIMER and DELAY TIMER parameters described later in this section.

The control station polls each active tributary every *n*th time it goes through the polling list (*n* is the active polling ratio set for that tributary). The default ratio is 1, meaning that unspecified active tributaries are polled each time the control station goes through the polling list. There is one dead ratio (default = 8) for all tributaries on a line. The control station polls one dead tributary each *x* times through the polling list (in round robin fashion). The control station polls a dying tributary four times as often as a dead one (or one-fourth the dead polling ratio).

For example, using the defaults, an active tributary is polled once a second, a dying tributary once every two seconds, and one of the dead tributaries every eight seconds. The exact frequency at which a particular tributary is polled depends not only on its active ratio, but also on the states and active ratios of other tributaries on the circuit and the amount of traffic.

To control which tributaries are active, set the related circuit state to ON or OFF. Note that dead tributaries are always polled if they are on. To improve performance for active tributaries, it is best to turn dead tributaries off if they will be dead for a significant period of time. You must also use the TRIBUTARY parameter to specify an address to be used by the polling mechanism. Correspondingly, you must set the same tributary address for that circuit on the remote node.

2.5.8.2 DMP and DMV Multipoint Controllers — For DMP and DMV controllers, you set the polling rates by using the DEAD TIMER and DELAY TIMER parameters in the SET LINE command (see Chapter 6).

The DEAD TIMER parameter specifies the polling rate for a dead tributary. Start by setting the DEAD TIMER parameter to a value midway between 5000 and 30,000; then adjust the rate to suit your network. If there are no problems with the rate you have set, you may wish to set a higher rate to maximize performance.

The DELAY TIMER, which specifies the maximum time to wait between polls in order to limit the effect of a fast control station on a slow tributary, can have any of the following values:

0	for lines with speeds equal to or greater than 56Kb
50	for multipoint EIA lines with speeds up to 19.2Kb
200	if any DDCMP software implementations are on the line

Always choose the larger value if more than one could apply.

2.5.9 Ethernet Circuit Parameters

There are two routing parameters that can be specified for Ethernet circuits in the DEFINE CIRCUIT command.

ROUTER PRIORITY — When two or more routing nodes are connected to an Ethernet cable, the routing node with the highest priority (a value in the range of 0 to 127) becomes the router designated to provide routing services for the end nodes on the Ethernet. (End nodes on the same Ethernet cable can communicate directly with one another; routers are required to send messages to nodes that are not connected to the Ethernet.) If two nodes share the highest priority, the node with the highest node address is selected. To learn which router is the designated router, issue a SHOW CIRCUIT command.

Example:

```
NCP>SHOW CIRCUIT UNA-0 CHARACTERISTICS

Circuit characteristics as of 3-OCT-83 15:03:27

Circuit = UNA-0

   Cost = 1, owner = XPT
   Type = Ethernet
   Designated router = 224 (ROBIN)
   Router priority = 70
```

MAXIMUM BROADCAST ROUTERS — This parameter specifies the maximum number of routers (other than the executor node) to be allowed on the identified Ethernet circuit. The practical maximum number of routers is about 10 because of the control traffic overhead (composed of routing messages and hello messages). The absolute maximum number of routers is 32.

2.5.10 DLM Circuit Parameters

Several circuit parameters that are specific to the operation of DLM circuits are described in the following sections.

2.5.10.1 Remote DTE Addresses — To establish an SVC with a remote DTE, the routing layer of DECnet software requires the address of the remote DTE. Use the NUMBER parameter in the DEFINE CIRCUIT command to specify the remote DTE address for an outgoing DLM SVC. The subaddress that you specify as the last number(s) of the DTE address must fall into the subaddresses range that is specified for the target node (see Section 2.1.3.2).

2.5.10.2 Re-calls for DLM Circuits — If an attempt to establish a DLM SVC has been unsuccessful, DECnet-RSX attempts to re-call the number. The RECALL TIMER parameter sets the interval that DECnet should wait before attempting a re-call (default: 30 seconds) and the MAXIMUM RECALLS parameter specifies the maximum number of times that DECnet should attempt a re-call (default: 5). You can modify either of these values in a DEFINE CIRCUIT command (see Chapter 6).

If the defined maximum re-call number is exceeded, the circuit is placed in the ON-FAILED state, and you must execute a SET CIRCUIT STATE ON command before another outgoing call can be attempted.

2.5.11 PSI Circuit Parameters

If you configured a PSI capability into your system, you can modify certain information that you specified during network generation by using the DEFINE CIRCUIT command with the parameters listed for PSI circuits in Table 2-6. If you use this command to modify the association of a PVC with a local DTE, the same DTE address must be specified in the protocol module (see DEFINE MODULE X25-PROTOCOL in Chapter 6). Note that both the packet and window size parameters were set to defaults for the PSN to which the circuit's DTE is connected. Consult the *RSX-11 PSI Network-specific Information* manual for details of these values before you modify them.

PSI channels. All virtual circuits for a DTE are multiplexed over the physical link between the DTE and the network interface (DCE). Each virtual circuit has a logical channel over which data is transmitted at both the DTE and DCE interfaces. Each channel is identified by a unique reference number called a logical channel number (LCN). Each DTE assigns a channel to the virtual circuit and recognizes the virtual circuit only by the LCN that identifies that channel. You use the CHANNELS parameter of the DEFINE MODULE X25-PROTOCOL command to associate the LCN with the DTE (see Chapter 6).

During network generation, you assigned a channel to each PVC. To modify the logical channel number associated with a circuit in the permanent data base, use the CHANNEL parameter in the DEFINE CIRCUIT command. You should not specify an LCN that has already been assigned to an outgoing call in the protocol module. PVC LCNs are assigned by the network vendor when the PVC is set up.

2.5.12 Circuit Counters

DECnet-RSX automatically maintains certain statistics, called counters, for circuits in the network. For all circuits, counter information can include the number of messages sent and received over the circuit, timeouts, and the amount of time since the counters were last zeroed. For DDCMP circuits, counters are maintained for timeouts and for data and buffer errors. For both Ethernet and DDCMP circuits, the number of bytes and data blocks sent and received are recorded. For PSI circuits, the following statistics are indicated in counters: the time since the counters were zeroed; the number of bytes, data blocks, and resets sent and received; the number of resets initiated by the network.

For more information on using counters, see Section 2.7. For a complete listing of circuit counters, see Appendix E.

2.6 Logging

DECnet-RSX software logs certain network-related events that occur during network operation. An event is defined as a network or system-specific occurrence for which the event logger maintains a record. Some events that can be recorded are listed below. A complete list is provided in Appendix D.

- Changes in line, circuit, node, and module states
- Lost events (events that were not logged)
- Service requests (when a line is put in an automatic service state)
- Passive loopback (when the executor is looping back circuit or line level test messages)
- Line, circuit, node, and module counter activity

This information can be useful in maintaining and tuning the network because it can be recorded continuously by the event logger. Section 3.3.2 provides an example of a typical event-logging situation with multiple nodes.

As system manager, you are responsible for controlling various aspects of event logging. For details on CFE, NCP, and VNP logging commands, see the following command descriptions in Chapter 6:

CFE Commands	NCP/VNP Commands
DEFINE LOGGING	SET LOGGING
PURGE LOGGING	CLEAR LOGGING
LIST LOGGING	SHOW LOGGING

Table 2-8 lists all logging parameters by function and groups them according to operational categories.

Table 2-8: Logging Parameters and Their Functions

Parameter Function	Event Source Parameters		Logging Component Parameters	
Event specification	EVENTS *event-list* KNOWN EVENTS			
Source of events	CIRCUIT			
	LINE			
	MODULE	X25-PROTOCOL X25-SERVER X29-SERVER		
	NODE			
Location to log events			SINK	EXECUTOR NODE *node-id* NODE $HOST
Logging component name			NAME	
State of logging component on executor node			STATE	OFF ON

Following network generation, all events generated by any component known to the local node are automatically logged at the executor console. You can use the SET LOGGING command to specify that only certain events be logged for certain components: a specified node, line, circuit, or module (X25-PROTOCOL, X25-SERVER, or X29-SERVER).

To remove any or all parameters from the volatile data base, use the CLEAR LOGGING command.

2.6.1 Event Specification

Events are defined by class and type. For the most part, events are logged for the various DNA layers and for system-specific resources. You can specify the kinds of events to be logged by using the following event list format with the EVENTS parameter:

class.type

where

class identifies the DNA layer or system-specific resource to which the event pertains.

type identifies a particular form of event, unique within an event class.

For example, events in the Transport layer are assigned to class 4. The event types for this class range from 0 through 19. Event type 0 indicates aged packet loss, event type 1 indicates unreachable node packet loss, and so forth (see Appendix D for a complete listing of events by class and type). Use the EVENTS parameter to specify a list of events to be logged. To specify logging of all DNA event classes and types that can be generated by a DECnet-RSX node, use the KNOWN EVENTS parameter. When removing logging parameters, use either of the above qualifiers, or specify ALL EVENTS when you want to clear not only all DNA events but also all user-defined events.

When providing an event list for the EVENTS parameter, you can specify only one class, but you can specify multiple event types within a class. You can specify a single event type, a range of types, a combination of these, or a wildcard character (*). The following examples illustrate the format of each.

Event List	Meaning
4.4	Specifies event class 4, type 4.
4.5-7	Specifies event class 4, types 5 through 7.
4.5,7-9,11	Specifies event class 4, types 5, 7 through 9, and 11. Note that types must be specified in ascending order.

2.6.2 Event Logger Components

Events can be logged on any of the following components: a console, a file, and/or a monitor program (as determined at network generation). A console is any record-oriented device that receives events in ASCII format. A file is a user-specified file that receives events in a DNA machine-readable binary

format. A monitor program is a system- or user-supplied program that receives and processes those events. Refer to the *DNA Network Management Functional Specification* for a description of the standard DNA event format. Appendix C describes the DECnet-RSX event-logging interface. This information is useful if you want to write a monitor program to receive events.

One or more of the components listed above can log events on your DECnet system. You can also have the event logger send events to logging components on remote DECnet nodes by using the SINK parameter (see SET LOGGING in Chapter 6).

When you initially specify components to which event data is to be logged, you can identify them by using the NAME parameter in the SET LOGGING command. Use the CLEAR LOGGING NAME command to clear the name of the logging device. This causes event logging to be sent to the default logging device (defaults are listed under SET LOGGING in Chapter 6).

2.7 Counters

DECnet-RSX software automatically maintains certain statistics, called counters, for the following network components:

- Circuits
- Lines
- Modules (X.25 Protocol and X.25 Server)
- Nodes (including the executor)
- System

All counters are listed with a brief description in Appendix E.

Counters are maintained on the node presently designated as the executor and can include such information as the number of data packets sent, received, or lost over a line; system buffer allocation failures; Transport packet information; or X.25 protocol message information. Counter statistics are useful alone or when read in conjunction with logging information (see Section 2.6 and Appendix D) to measure and evaluate the performance and throughput of your network configuration.

As system manager, you can display counters (see Section 2.7.1) and periodically zero them (see Section 2.7.2) using the NCP SHOW and ZERO commands. In addition, for PSI circuits, lines, and modules, you can set a timer whose expiration causes counters for a given component to be logged as an event (see Section 2.7.3). All network management commands for manipulating counters are described in detail in Chapter 6.

2.7.1 Displaying Counters

You can use the NCP SHOW command (see Chapter 6) to display counters for any of the network components listed at the beginning of Section 2.7. A sample command and resulting display are shown below:

```
NCP>SHOW CIRCUIT DMC-0 COUNTERS


Circuit counters as of 23-OCT-83 15:18

Circuit = DMC-0

     >65534  Seconds since last zeroed
       1568  Terminating packets received
       1753  Originating packets sent
    1467600  Transit packets received
    1544950  Transit packets sent
          3  Transit congestion loss
         15  Circuit down
          0  Initialization failure
  102139870  Bytes received
   90378851  Bytes sent
    1606602  Data blocks received
    2023269  Data blocks sent
          0  Data errors inbound
          0  Data errors outbound
          0  Remote reply timeouts
          0  Local reply timeouts
          0  Local buffer errors
```

Some of these counters can be qualified by information that indicates the condition that contributed to the error.

To gain a detailed understanding of counters or the software design and algorithms they represent, consult the appropriate architectual specifications. Refer to Chapter 3 for additional information on the NCP SHOW commands that display network status and counter information.

Several counters in the display above correspond to network logging events. The events and event descriptions may provide additional information relative to the specific occurrence. Refer to Appendix D for a complete description of events that can be logged. Appendix E maps individual counters to corresponding events where applicable.

2.7.2 Zeroing Counters

When the network is running, you can use the NCP ZERO command (see Chapter 6) to zero any of the network counters. It is wise to zero counters periodically or they will eventually exceed (overflow) their maximum count. Counters that have overflowed display a "greater than" sign (>) in front of their maximum count (see the sample display, above).

For each component there is a special counter that indicates the number of seconds that have elapsed since the counters for that component were last zeroed. The software increments this counter every second and clears it when other counters for the component are cleared.

2.7.3 Counter Timers and Logging Counters

For PSI lines, PSI circuits, and PSI modules, you can use network management commands to affect the frequency with which counters are logged and then automatically zeroed. To set a timer whose expiration automatically causes counters to be logged on the logging component and then to be zeroed, use the COUNTER TIMER parameter with an NCP/VNP SET CIRCUIT, SET LINE, or SET MODULE command (see Chapter 6). For example,

```
NCP>SET LINE KMX-0-0 COUNTER TIMER 600
```

causes a line counter logging event to take place every 600 seconds for line KMX-0-0. To clear this parameter, use the following NCP command:

```
NCP>CLEAR LINE KMX-0-0 COUNTER TIMER
```

This command zeroes the counter timer in the volatile data base. You can also set and clear counter timers in the permanent data base by using the CFE DEFINE LINE and PURGE LINE commands.

When counters are logged for a particular component as the result of a counter timer expiration, event 0.8 is generated, followed by a listing of the counters.

Example:

```
 Event type 0.8 Automatic counters
Occurred 6-NOV-83 13:17:43 on node 56 (BLTMRE)
Circuit QTPVC
22  Seconds since last zeroed
0  Bytes received
0  Bytes sent
0  Data block received
0  Data blocks sent
0  Locally initiated resets
0  Remotely initiated resets
0  Network initiated resets
```

2.8 Processes

DECnet-RSX processes provide specific functions for the DECnet-RSX system. For example, the ECL process controls logical link handling, while NW controls X.25 level 3 protocol operations. There are also device driver processes that control communications devices for the system. The RSX operating system is unaware of these processes, except as residents of partitions in system memory.

The following network management commands (described in detail in Chapter 6) can be used to manipulate DECnet-RSX processes.

CFE Commands	NCP/VNP Commands
DEFINE PROCESS	SET PROCESS
—	CLEAR PROCESS
LIST PROCESS	SHOW PROCESS

Table 2-9 lists the parameters that can be specified on DEFINE and SET commands.

Table 2-9: Process Parameters

Parameter	DEFINE	SET
ALL		X
LOCATION		X
MAXIMUM CONTROLLERS *count*	X	X
MAXIMUM LINES *number*	X	X
PARTITION *partition-name*		X
STATE	X	

2.8.1 Process Identification

You can use network management commands to modify parameters for an individual process or for all known processes. Use the KNOWN PROCESSES keyword to specify all processes identified to the system during network generation. To specify an individual process, use a process name from Table 2-10. Table 2-10 lists processes alphabetically by type, including device driver processes and software protocol-related processes for DECnet and PSI.

2.8.2 Process States and Loading

During network generation, the required processes are set up in the permanent data base and are automatically loaded when you execute an NCP SET SYSTEM command. Likewise, processes for lines and circuits are automatically loaded when you issue one of the following commands:

CFE>DEFINE LINE *line-id* STATE ON

NCP>SET LINE *line-id* ALL

A process remains in memory until you clear the system or the process.

To load a process that has not been automatically loaded into the volatile data base, use the NCP SET PROCESS ALL command. This loads the identified process with the default information specified in the permanent data base for each process parameter.

The LOCATION parameter allows you to specify the memory location into which to load each process when it is manually loaded into the volatile data base. This enables you to manage memory allocation for the system.

2.8.3 Maximum Controllers and Lines

During network generation, you specify the maximum number of lines and controllers that each process can control. If necessary, you can use the MAXIMUM CONTROLLERS and MAXIMUM LINES parameters to modify this information. These parameters determine the amount of additional space to be reserved in the process partition that is used to allocate process tables. If more lines are loaded than specified by the MAXIMUM LINES parameter, the additional line tables are allocated from the system pool.

Table 2-10: DECnet-RSX Processes

Process Name	Function
Communications Executive processes	
AUX	Auxiliary process
DLX	Direct Line Access Controller
EVL	Event Logger process
DECnet device drivers	
DL	DL-11 driver
DLV	DLV-11 driver
DMC	DMC-11/DMR-11 driver
DMP	DMP-11 driver
DMV	DMV-11 driver
DPV	DPV-11 driver
DU	DU-11 driver
DUP	DUP-11 driver
DUV	DUV-11 driver
DV	DV-11 driver
DZ	DZ-11 driver
DZV	DZV-11 driver
KDP (KMC/DUP)	KDP-11 driver
KDZ (KMC/DZ)	KDZ-11 driver
PCL	PCL driver
UNA	DEUNA driver
QNA	DEQNA driver
DECnet processes	
DCP	Implements DDCMP; included for all systems with devices other than DMC-11, DMP-11, DMR-11, DMV-11, PCL-11B, DEUNA, and DEQNA
EPM	Ethernet Protocol Manager; included for all systems with Ethernet devices
ECL	End Communications layer process
XPT	Transport layer process
PSI device drivers	
KMX	KMX-11 driver
SDP	DUP-11 driver
SDV	DPV-11 driver
PSI processes	
DLM	Data-link-mapping process
LAB	X.25 level 2 LAPB protocol process
NW	PSI user interface process
PLI	X.25 level 3 (packet level) process

2.9 PSI Modules

There are four PSI modules that can be manipulated by DECnet-RSX network management commands:

- **X.25 protocol module.** Identifies local DTE addresses and group names
- **X.25 server module and X.29 server module.** Identifies destinations
- **X.25 access module.** Replaces remote DTE addresses with logical destination names

Each module, its function, and its parameters are briefly described in the following sections. For full details, see the applicable SET/DEFINE, CLEAR/PURGE, and SHOW/LIST command descriptions in Chapter 6.

2.9.1 X.25 Protocol Module

The X.25 protocol module implements the X.25 level 3 protocol that controls the transmission of data packets. In particular, this module structures control and user data into packets; sequences these packets for transmission; and establishes, maintains, and clears PSI virtual circuits. This module also associates local DTE addresses and group names with this controlling information.

Table 2-11 lists the three types of X.25 protocol parameters that can be specified in DEFINE and SET commands. Each type is briefly discussed in the following sections. For more information on parameters for DEFINE/SET MODULE X25-PROTOCOL commands, see Chapter 6.

2.9.1.1 Protocol-related Parameters — During network generation, you specified a number of protocol-related parameters that affect the operation of the X.25 protocol module. These parameters can be modified only in the permanent data base.

Both the packet and window size parameters were set to defaults for the PSN to which the DTE is connected. Consult the *RSX-11 PSI Network-specific Information* manual for details of these values before modifying them.

When you use the MAXIMUM DATA parameter to specify the maximum size of a packet for all SVCs, specify a size at least 5 bytes smaller than the maximum size specified for a frame on a line (see DEFINE LINE in Chapter 6). This value must not be exceeded when you specify a companion value for the DEFAULT DATA parameter. For complete descriptions of all protocol-related parameters, see Chapter 6.

2.9.1.2 Local DTE-related Parameters — Every DTE in the network is identified by one or more local DTE addresses. You can use network management commands to modify parameters for a specific DTE or for all known DTEs. Use the KNOWN DTES keyword to modify parameters for all local DTEs identified to the system during network generation. To modify parameters for a specific DTE, use the DTE keyword with an integer address consisting of 1 to 15 digits.

Table 2-11: Protocol Module Parameters

Parameter	DEFINE	SET
Protocol-related parameters		
MAXIMUM DATA	X	
DEFAULT DATA	X	
MAXIMUM WINDOW	X	
DEFAULT WINDOW	X	
CALL TIMER	X	
CLEAR TIMER	X	
MAXIMUM CLEARS	X	
RESET TIMER	X	
MAXIMUM RESETS	X	
RESTART TIMER	X	
MAXIMUM RESTARTS	X	
Local DTE-related parameters		
DTE	X	X
COUNTER TIMER	X	X
STATE	X	X
LINE	X	
MAXIMUM CIRCUITS	X	
CHANNELS	X	
Group-related parameters		
GROUP	X	X
DTE	X	X
NUMBER	X	X
TYPE BILATERAL	X	X

2.9.1.3 Group-related Parameters — At subscription time, you may have chosen an optional user group facility to restrict DTE communication. You can use network management commands to modify parameters for a specific group or for all known groups. Use the KNOWN GROUPS keyword to modify parameters for all groups identified to the system during network generation. To modify parameters for a specific group, use the GROUP keyword with a group name consisting of 1 to 6 alphanumeric characters. Every group name must in turn be associated with a local DTE address. Use the NUMBER parameter to modify the group number allocated at subscription time.

2.9.2 X.25 Server and X.29 Server Modules

Whenever a remote DTE attempts to communicate with your local DTE (that is, attempts to set up a virtual circuit), both the remote DTE and the system manager at the local DTE provide information that identifies the destination of the call. This information consists of the remote DTE address, a local DTE address, possibly a closed user group (CUG) or bilateral closed user group (BCUG) name, and possibly a value in the user data field. DECnet-RSX software uses this information to determine how to handle the incoming call to the local DTE.

How the software handles incoming calls depends upon the parameters you define for X.25 server and X.29 server module destinations. The software uses X.25 server module parameters to handle incoming calls for the PSI user program interface. It uses X.29 server module parameters to handle incoming calls for the PSI X.29 terminal interface. For information on incoming call handling for DECnet-RSX with a PSI capability, see the *RSX-11 PSI System Manager's Guide.*

Server module parameters consist of two types: server-related parameters, which affect the operation of the server software, and destination-related parameters, which define information for handling incoming calls.

Table 2-12 lists by type all server module parameters that can be specified in DEFINE and SET commands. For more information, see DEFINE/SET command descriptions in Chapter 6.

Table 2-12: Server Module Parameters

Parameter	DEFINE	SET
Server-related parameters		
COUNTER TIMER	X	X
STATE (1)		X
MAXIMUM CIRCUITS	X	
Destination-related parameters		
DESTINATION	X	X
CALL MASK	X	X
CALL VALUE (2)	X	X
GROUP (3)	X	X
NUMBER	X	X
SUBADDRESSES	X	X
OBJECT (4)	X	X
PRIORITY	X	X

1. You cannot specify STATE for the X.29 server module.
2. A value of 01 is recommended for the X.29 server module.
3. This group name must match the one specified in the protocol module.
4. Specify a task number for X.25 multicopy tasks and a task name for X.29 destinations.

2.9.2.1 Destination-related Parameters — You can use network management commands to modify parameters for a specific destination or for all known destinations. Use the KNOWN DESTINATIONS keyword to modify parameters for all destinations identified to the system during network generation. To modify parameters for a specific destination, use the DESTINATION keyword with a destination name consisting of 1 to 6 alphanumeric characters.

There are five destination parameters that you can modify to set up incoming call specifications. Two other parameters specify the priority of the destination and the task (object) to which the destination passes the call for processing.

NOTE

> You cannot modify parameters for existing destinations. You can only create new destinations. Existing destinations can be cleared, then reset with new parameters. If you specify a destination name, you must also specify the object.

2.9.2.2 Server-related Parameters — During network generation, you specified the maximum number of circuits that each server module could handle simultaneously (including both incoming and outgoing calls). This count can be modified in the permanent data base by specifying the MAXIMUM CIRCUITS parameter.

In the volatile data base, you can set the X.25 server module state to one of the following values:

OFF — Allows no virtual circuit activity, terminates existing circuits, and forces the release of all mailboxes immediately.

ON — Allows normal virtual circuit activity.

SHUT — Allows no new virtual circuits, does not destroy existing virtual circuits, and goes to the OFF state when all virtual circuits terminate and all mailboxes are released.

2.9.3 X.25 Access Module

You can use the X.25 access module to create PSI logical destination names to use like alias node names (see Section 2.1.1.2) in place of a remote DTE address whenever you identify a remote DTE in a PSN. RSX-11 PSI software handles the translation from logical destination name to remote DTE address. You can create or modify logical destination names in any of the three data bases. For example, to create a logical destination name in the volatile data base, use the NCP SET MODULE X25-ACCESS command:

```
NCP>SET MODULE X25-ACCESS DESTINATION STEVE NUMBER 12356 TERMINAL TT11:
```

This command specifies that logical destination name STEVE can be used in place of remote DTE number 12356 for network user programs initiated from terminal TT11:. You can specify whether the logical destination name applies to a specific terminal (TERMINAL *term-id*), as in this example, or to all terminals (GLOBAL). The default scope is your terminal.

See the SET/DEFINE, CLEAR/PURGE, and SHOW/LIST MODULE X25-ACCESS command descriptions in Chapter 6 for full details on parameters and usage.

2.9.4 PSI Module Counters

RSX-11 PSI automatically maintains certain statistics, called counters, in the network for the X.25 protocol, X.25 server, and X.29 server modules. For the X.25 protocol module, these counters include the number of bytes, data blocks, calls, and fast selects sent and received; the number of active channels; the number of resets sent, received, or initiated by the network; and the number of restarts. These statistics are useful in monitoring the activity of the component. X.25/X.29 server module counters include the maximum number of active circuits, the number of incoming calls rejected, the time since the counter was last zeroed, and a counter timer value. See Appendix E for a complete list of all PSI module counters.

Chapter 3
Operating DECnet-RSX/PSI Nodes

This chapter provides the basic information you need to perform the following functions:

- Start up and shut down a local DECnet-RSX node (see Section 3.1)
- Start up and shut down a local DECnet-RSX/PSI node (see Section 3.2)
- Monitor local and remote nodes within a DECnet-RSX/PSI network (see Section 3.3)
- Test network operation (see Section 3.4)

Specific CFE, NCP, and VNP commands are referenced throughout this chapter. See Chapter 6 for more information on using these commands and their parameters.

3.1 Starting Up/Shutting Down a Local DECnet-RSX Node

3.1.1 Starting Up a Local DECnet-RSX Node

The start-up procedure for a local node loads the Communications Executive (CEX) into memory and activates the DECnet-RSX software. If your DECnet system includes PSI, PSI is automatically loaded, but you must still start it up (see Section 3.2.1).

To load the DECnet software into the volatile data base, use one of the following methods:

- The NETINS.CMD file (for newly generated systems)
- NCP commands (for previously installed software)
- VNP commands (to reboot from the system image file; not supported by DECnet-11M-PLUS)

For more information on using the NCP and VNP utilities, see Chapter 1.

3.1.1.1 Using the NETINS.CMD File — The NETINS.CMD file is a product of NETGEN and is described in detail in the *DECnet-RSX Network Generation and Installation Guide*. After running NETGEN, you must use NETINS.CMD to start up the newly generated DECnet-RSX system for the first time. To load a DECnet-RSX system using this file, enter

```
@[xxx,1]NETINS
```

where *xxx* is the network UIC group code selected at NETGEN.

3.1.1.2 Using NCP Commands — Use the following NCP commands to load and start up a DECnet-RSX system when the network software has already been installed (by a previous run of NETINS.CMD, for example).

```
NCP>SET SYSTEM
NCP>SET EXECUTOR STATE ON
```

3.1.1.3 Using VNP Commands — Use the following VNP commands to incorporate the DECnet-RSX software components into the RSX-11M or RSX-11S system image file (RSX-11M-PLUS does not support VNP). You must perform an RSX SAV on the system image file before you run VNP. Once you have run VNP, DECnet-RSX will start up automatically the next time you boot the RSX operating system.

```
>VNP
Enter filename: RSX11S.SYS        (or RSX11M.SYS)
VNP>SET SYSTEM
VNP>SET EXECUTOR STATE ON
```

3.1.2 Loading/Turning On a Line in a Running System

After you have started up the system, you can still load a line (or all lines known to the system) and turn on the associated circuit(s) by issuing one of the following pairs of commands:

```
NCP>SET LINE line-id ALL
NCP>SET CIRCUIT circuit-id STATE ON

NCP>SET KNOWN LINES ALL
NCP>SET KNOWN CIRCUITS STATE ON
```

All lines to be loaded must be in the CLEARED state. The ALL parameter specifies that the line(s) are to be loaded with the default parameter values specified in the permanent data base. If you wish to change the defaults, you must specify the appropriate SET LINE command parameters (see Chapter 6 for details).

To cause a new line (or all known lines) to be loaded and turned on the next time the system is started up, use the CFE DEFINE versions of the commands shown above to set up the line(s) and circuit(s) in the permanent data base. The line(s) will be automatically loaded and their circuit(s) turned on the next time you issue the SET SYSTEM and SET EXECUTOR STATE ON commands.

3.1.3 Shutting Down DECnet-RSX

You can shut down selected lines or the entire node, as described in the following sections.

3.1.3.1 Shutting Down DECnet-RSX Lines — You shut down a DECnet line by setting the state of the associated circuit to OFF. You can shut down one or all DECnet lines by using one of the following commands:

```
NCP>SET CIRCUIT circuit-id STATE OFF

NCP>SET KNOWN CIRCUITS STATE OFF
```

3.1.3.2 Shutting Down a DECnet-RSX Node — Use the NCP SET EXECUTOR STATE command to shut down node activity immediately or in an orderly fashion, depending upon the state you specify.

State	Result
SHUT	**Orderly shutdown.** Prevents any new logical links to or from the node, but does not cut off currently active links. After all links have disconnected, the system returns event message 2.0, stating that the executor state is OFF.
OFF	**Immediate shutdown.** Immediately terminates all network activity without allowing active links to disconnect in an orderly manner.

You can also initiate the orderly shutdown of a DECnet-RSX node by executing the following NETGEN output command file:

```
>@[xxx,1]NETREM
```

where *xxx* is the network UIC group code selected at NETGEN. For more information, see the *DECnet-RSX Network Generation and Installation Guide*.

3.2 Starting Up/Shutting Down a Local DECnet-RSX/PSI Node

3.2.1 Starting Up a Local DECnet-RSX/PSI Node

The start-up procedure for a local node loads the Communications Executive (CEX) into memory and activates the DECnet-RSX software. PSI is automatically loaded, but you still have to start it up.

To load the DECnet software into the volatile data base, use one of the following methods:

- The NETINS.CMD file (for newly generated systems)
- NCP commands (for previously installed software)
- VNP commands (to reboot from the system image file; not supported by DECnet-11M-PLUS)

For more information on using the NCP and VNP utilities, see Chapter 1.

3.2.1.1 Using the NETINS.CMD File — The NETINS.CMD file is a product of NETGEN and is described in detail in the *DECnet-RSX Network Generation and Installation Guide*. After running NETGEN, you must use NETINS.CMD to start up a newly generated DECnet-RSX/PSI system for the first time. To load a DECnet-RSX/PSI system using this file, enter

```
>@[xxx,1]NETINS
```

where *xxx* is the network UIC group code selected at NETGEN.

3.2.1.2 Using NCP Commands — Use the following NCP commands to load and start up a DECnet-RSX/PSI system when the network software has already been installed (by a previous run of NETINS.CMD, for example).

```
NCP>SET SYSTEM
NCP>SET EXECUTOR STATE ON
NCP>SET MODULE X25-SERVER STATE ON
```

3.2.1.3 Using VNP Commands — Use the following VNP commands to incorporate the DECnet-RSX/PSI software components into the RSX-11M or RSX-11S system image file (RSX-11M-PLUS does not support VNP). You must perform an RSX SAV on the system image file before you run VNP. Once you have run VNP, DECnet-RSX/PSI will start up automatically the next time you boot the RSX operating system.

```
>VNP
Enter filename: RSX11S.SYS        (or RSX11M.SYS)
VNP>SET SYSTEM
VNP>SET EXECUTOR STATE ON
VNP>SET MODULE X25-SERVER STATE ON
```

3.2.2 Loading/Turning On a PSI Line in a Running System

After you have started up the system, you can still load and turn on a specific PSI line or all PSI lines known to the system by issuing one of the following commands:

```
NCP>SET LINE line-id ALL STATE ON

NCP>SET KNOWN LINES ALL STATE ON
```

All lines to be loaded must be in the CLEARED state. The ALL parameter specifies that the line(s) are to be loaded with the default parameter values. If you wish to change the defaults, you must specify the appropriate SET LINE command parameters (see Chapter 6 for details).

To cause a new line (or all known lines) to be loaded and turned on the next time the system is started up, use the CFE DEFINE versions of the commands shown above to set up the line(s) in the permanent data base. The line(s) will be automatically loaded and turned on the next time you issue the SET SYSTEM and SET MODULE X25-SERVER STATE ON commands.

3.2.3 Shutting Down PSI

You can shut down selected PSI components or the whole PSI module, as described in the following sections.

3.2.3.1 Shutting Down PSI Components — To shut down one or all PSI lines, use one of the following commands:

```
NCP>SET LINE line-id STATE OFF
```

```
NCP>SET KNOWN LINES STATE OFF
```

To shut down one or all DTEs, use one of the following commands:

```
NCP>SET MODULE X25-PROTOCOL DTE dte-address STATE SHUT
```

```
NCP>SET MODULE X25-PROTOCOL KNOWN DTES STATE SHUT
```

The system will eventually return an event message saying that the specified DTE(s) have been shut down.

3.2.3.2 Shutting Down a PSI Module — Use the NCP SET MODULE X25-SERVER STATE command to shut down PSI immediately or in an orderly fashion, depending upon the state you specify.

State	Result
SHUT	**Orderly shutdown.** Prevents any new virtual circuit connections to the module, but does not cut off current connections. After all virtual circuits have disconnected, the system returns an event message stating that the module state is OFF.
OFF	**Immediate shutdown.** Immediately terminates all virtual circuit connections without allowing circuits to disconnect in an orderly manner.

When PSI operation has ceased, you can then shut down DECnet-RSX on the local node (see Section 3.1.3.2).

3.3 Monitoring DECnet-RSX/PSI Nodes

DECnet provides several means of monitoring network activity once you have brought up the local DECnet system.

- You can use CFE, NCP, or VNP commands to display information about network components.
- You can use the Event Logger (EVL) to log dynamic network events.
- You can use the Network Display Program (NTD) to display reachable nodes or a specific node's resources.

This section provides an overview of each of these facilities.

3.3.1 NCP, VNP, and CFE Display Commands

NCP, VNP, and CFE provide commands that enable you to display information about network components on your terminal.

- The NCP SHOW command displays component information from the volatile data base. For example, if a line failure or a change of state on a line or circuit occurs on a routing node, you can issue an NCP SHOW ACTIVE NODES command to determine which known nodes are now reachable. Depending on the component specified, you can select from the following types of display:

  CHARACTERISTICS displays static information about the component, such as the local node identification and relevant routing parameters (for the EXECUTOR component) or the names and numbers of known network objects.

  COUNTERS provides counter information for circuits, modules, lines, nodes, and the system. (See Chapter 2 for a discussion of counters.)

  EVENTS (SHOW LOGGING only) displays information about events currently being logged.

  STATUS shows dynamic information that usually reflects network activity for the running network. Depending on the component, this can include the local node and its operational state, reachable and unreachable nodes, the local DTE and its operational state, and circuits and lines and their operational states.

  SUMMARY (default) includes only the most useful information derived from both static and dynamic sources. This is usually an abbreviated list of information provided for both the CHARACTERISTICS and STATUS display types.

- The VNP SHOW command displays component information from the system image file. Depending on the component, you can choose a CHARACTERISTICS, EVENTS, or SUMMARY (default) display type. See the definitions of these display types given above.

- The CFE LIST command automatically displays static information for the specified component (see the description of CHARACTERISTICS above).

Chapter 6 describes the display commands for all three utilities.

3.3.1.1 Network Components — The components for which you can display information are:

- Aliases
- Circuits
- Lines
- Logging
- Modules
- Nodes, including the executor node
- Objects
- Processes
- System
- Trace

In most cases you can display information for one particular component (for example, for LINE DMC-0) or for all components of the specified type that fall within one of several group classifications. The group classifications that you can specify are:

- KNOWN — Displays information for all instances of a component available to the local node; for example,

  ```
  NCP>SHOW KNOWN NODES CHARACTERISTICS
  ```

- SIGNIFICANT — Shows information about all instances of a known component for which information is available; for example,

  ```
  NCP>SHOW SIGNIFICANT LINES
  ```

- ACTIVE — Displays information for all active components (that is, those components whose state is other than OFF or CLEARED); for example,

  ```
  NCP>SHOW ACTIVE LINES CHARACTERISTICS
  ```

- ALL (valid only for NCP/VNP SHOW ALIAS and SHOW MODULE X25-ACCESS commands) — Displays information for all aliases or logical destination names regardless of scope; for example,

  ```
  NCP>SHOW ALL ALIASES
  ```

See SHOW command descriptions in Chapter 6 to determine which classifications apply to each component.

3.3.1.2 Copying NCP Display Information to a File — All NCP display commands optionally allow you to direct the display information to a user-specified output file instead of displaying it on your terminal. If the specified file already exists, NCP appends the display information to the file.

Example:

```
NCP>SHOW KNOWN LOGGING TO NET.LOG
```

This example creates the file NET.LOG (under the current UIC), which will contain current summary information for all known logging on the running network.

3.3.1.3 NCP SHOW Command Examples — The following examples illustrate various NCP SHOW command displays.

Example 1:

```
NCP>SHOW EXECUTOR CHARACTERISTICS

Node characteristics as of 13-SEP-83 16:29:30

Executor node = 19 (BURGER)

   Identification = DISTRIBUTED SYSTEMS, Management version = 4.0.0
   Host = 19 (BURGER), Loop count 1, Loop length = 40
   Loop with = Mixed, NSP version = 4.0.0
   Maximum links = 20, Routing version = 2.0.0, Type = Routing IV
   Routing timer = 300
   Maximum address = 300, Maximum circuits = 8
   Maximum cost = 1022
   Maximum hops = 10, Maximum visits = 20
   Maximum broadcast nonrouters = 32
   Maximum broadcast routers = 15
   Transmit password = ...
   Verification state = On
```

Example 2:

```
NCP>SHOW KNOWN LINES

Known lines summary as of 13-SEP-83 16:32:34

Line               State

DMC-0              On
DMC-1              Cleared
PCL-0              On
UNA-0              On
UNA-1              Cleared
```

Example 3:

```
NCP>SHOW LINE UNA-0 CHARACTERISTICS

Line characteristics as of 13-SEP-83 16:38:06

Line = UNA-0

    Controller = Normal
    Protocol = Ethernet
    Hardware address = AA-00-03-00-01-13
    Controller CSR = 174510, Vector = 120, Priority = 5
```

Example 4:

```
NCP>SHOW LINE PCL-0 COUNTERS

Line = PCL-0

        18820  Seconds since last zeroed
            0  Attempts to become master
            0  Process errors
            0  Device errors
```

Example 5:

```
NCP>SHOW CIRCUIT PCL-0.2 CHARACTERISTICS

Circuit characteristics as of 14-SEP-83 10:02:47

Circuit = PCL-0.2

  Cost = 2, Hello timer = 15, Listen timer = 30
  Owner = XPT
  Type = DDCMP Controller, Tributary = 3
```

Example 6:

```
NCP>SHOW KNOWN OBJECTS

Known objects summary as of 13-SEP-83 16:39:18

Object   Name     Copies   User      Verification

0                 Single   Default   Off
15       TCL...   Single   Default   On
16       LSN$$$   5        Default   Off
17       FAL$$$   9        Login     On
18       HLD...   Single   Default   Off
19       NIC$$$   5        Default   Inspect
23       RMHACP   Single   Default   Off
25       MIR$$$   5        Default   Off
26       EVR...   Single   Default   Off
63       DTR...   Single   Default   Off
```

3.3.2 Event Logging

Event logging allows you to monitor selected network activity that might otherwise go unobserved. The events that can be logged include those occurring at the local node and at remote DTEs. Refer to Chapter 2 for a general overview of event logging.

The following network management commands control event logging (see Chapter 6 for details):

CFE Commands	NCP/VNP Commands
DEFINE LOGGING	SET LOGGING
PURGE LOGGING	CLEAR LOGGING
LIST LOGGING	SHOW LOGGING

A simple example of how you can use event-logging commands to monitor local and remote nodes in your network is shown below. This sequence of NCP commands sets up event logging for a local node called ALB and for two remote nodes called PHL and BOS. In response to the first command, ALB's system console will display all known events that occur locally. In response to the next four commands, remote nodes BOS and PHL will each log all known events locally at the system console and remotely at node ALB's system console.

```
NCP>SET LOGGING CONSOLE KNOWN EVENTS STATE ON
NCP>TELL BOS SET LOGGING CONSOLE KNOWN EVENTS STATE ON
NCP>TELL BOS SET LOGGING CONSOLE SINK NODE ALB KNOWN EVENTS
NCP>TELL PHL SET LOGGING CONSOLE KNOWN EVENTS STATE ON
NCP>TELL PHL SET LOGGING CONSOLE SINK NODE ALB KNOWN EVENTS
```

Appendix D describes the event message format and lists the text of all DECnet-RSX event messages.

3.3.3 The Network Display Program (NTD)

You can run the Network Display Program (NTD) on any RSX or IAS node in the network. NTD can produce two types of real-time displays:

- A list of all reachable nodes in the network
- A resource display for a specified node (provided the node supports NTDEMO, the task that collects the information that is displayed)

You can display this information on a local VT100 or VT52 terminal where it is continuously updated, or you can obtain a snapshot printout of the display information on a hard-copy terminal. Chapter 11 describes NTD in detail.

3.4 Testing Network Operation

This section discusses two types of tests:

- Loopback tests that exercise software and hardware components of a DECnet-RSX node
- A trace facility, composed of network management TRACE commands and a trace interpreter task (TRI), for diagnosing software problems on a PSI line between a local DTE and the DCE to which it is connected

3.4.1 Loopback Tests

DECnet-RSX loopback tests allow you to exercise hardware and software components to verify that the node is working properly. Several levels of tests are provided: node level, circuit level, and line level (for PSI lines). Node level tests verify the logical link capability of a node. Circuit level tests verify the software controlling a circuit. DECnet-RSX provides node and circuit level loopback tests to verify both DDCMP and Ethernet hardware and software. You might use these tests under the following circumstances:

- A system has just been installed.
- A system has been modified.
- You want to isolate a fault.
- A situation arises that causes you to doubt the integrity of the system.

Details of these tests are found in the *DECnet-RSX Postinstallation Checkout Procedures* manual. Chapter 6 describes the following network management commands that you can issue to run the tests:

NCP LOOP EXECUTOR
NCP LOOP NODE
NCP LOOP LINE
NCP LOOP CIRCUIT

3.4.2 The Trace Facility

The DECnet-RSX trace facility enables you to diagnose software problems on a PSI line. The facility traces message frames passing between the X.25 level 2 protocol and the device driver and copies the traced frames to a disk file. The NCP SET TRACE command initiates the tracing and specifies the file to contain the trace data. You then use the trace interpreter task (TRI) to analyze and print out the contents of the trace file. Chapter 13 describes the TRI command and its switches, and Chapter 6 describes the NCP SET, CLEAR, and SHOW TRACE commands.

Chapter 4
Supporting Remote RSX–11S Systems

This chapter describes procedures for supporting a remote RSX–11S system. These procedures include

- Down-line loading an RSX–11S system (Section 4.1)
- Up-line dumping of RSX–11S system memory (Section 4.2)
- Down-line loading, checkpointing, and overlaying of RSX–11S tasks (Section 4.3)

4.1 Down-line Loading RSX–11S Systems

Down-line loading is a function that allows a remote target node to receive an RSX–11S operating system file image from a host (executor) node. You can issue an NCP LOAD or NCP TRIGGER command to initiate the down-line load, or an operator at the target can request a load.

NOTE

Server software products that run on Digital Ethernet Communication Server (DECS) hardware, such as the DECnet Router Server and the Terminal Server, are described in separate documentation (see the Preface). The server-specific documentation contains complete instructions for down-line loading each server product from a host node. However, the procedures you use to load a server down line from a DECnet–RSX host are based on the procedures described in this chapter.

In DECnet–RSX, the executor and the host must be the same node when you are loading a system image down line. In other words, the executor node, which actually performs the load and is adjacent to the target, must have local access to the load files. However, for down-line task loading, the host node for the target can be different from the executor node that loaded it (see Sections 4.1.4.2 and 4.3).

The load sequence is the same, whether you initiate the load with NCP or an operator requests the load from the target. The primary loader is the first program to run at the target node. Typically, either this program is executed directly from the target node's bootstrap ROM, or it is included in the microcode of the load device (DMC, DMR, DMP, DMV, UNA, or QNA). Once the target node's primary loader is triggered, the target node sends a message to the executor node requesting a program load. Usually, the primary loader requests a secondary loader program, which in turn can request a tertiary loader. The final program to be loaded is the operating system. In this sequence, each program requests the next one until the operating system is loaded.

At the end of the load sequence, the executor sends a message to the target that includes the target's name and address and the identification of its host node (see Section 4.1.4). If the target system is a Phase IV RSX-11S system, the message also includes the date and time. The information obtained from the message is entered in the target system's data base.

This section provides basic information about preparing to down-line load and about using NCP commands to load the system.

- Section 4.1.1 describes setting up an RSX-11S system image file to be loaded down line to a target node.
- Section 4.1.2 describes various prerequisites for down-line loading.
- Section 4.1.3 describes the NCP commands you can use to initiate the down-line load.
- Section 4.1.4 describes the down-line loading data base and the parameters it contains.

4.1.1 Setting Up an RSX-11S System Image File

In order to down-line load an RSX-11S node, you must create a system image file that is either a DECnet-11S node or a stand alone RSX-11S system (a process control system, for example). To create a DECnet-11S node, use VMR and VNP to create the system image file. Note that if you want to load DECnet software above the 124KW boundary, you must ensure that the system image file is large enough to accommodate it. The system created by the following example has its circuit set to ON:

```
>VMR
Enter filename: RSX11S
VMR> SET /POOL=260
VMR> SET /MAIN=MCRPAR:260:63:TASK
VMR> SET /MAIN=CEXPAR:343:41:COM
VMR> INS BASMCR/FIX=YES
VMR> INS NETACP/CKP=NO/FIX=YES
VMR> INS NTINIT/FIX=YES
VMR> CTRL/Z
>VNP RSX11S
VNP> SET SYSTEM ALL
VNP> SET EXECUTOR STATE ON
VNP> SET LINE DLV-0 ALL
VNP> SET CIRCUIT DLV-0 STATE ON
VNP> EXIT
```

Note that an RSX-11S system does not have to be down-line loaded. Given the appropriate hardware, you can also load RSX-11S by booting it from a disk, a diskette, or a magnetic tape. When booting from a local load device, DECnet must be present in the booted image.

4.1.2 Set-up Requirements for Down-line Loading

Before attempting to down-line load, you must ensure that nodes, lines, and circuits meet the following requirements:

- The target node must be connected directly to the executor node. The executor node provides access to the line and circuit levels.
- The primary loader must be a cooperating program either in the target or in the microcode of the target's device (DMC, DMR, DMP, DMV, UNA, or QNA). The down-line load operation usually involves loading a series of bootstraps, each of which requests the next program until the operating system itself is loaded.
- To allow the tertiary loader to complete the load sequence, the RSX-11S system image itself must be 2K words less than the total memory on the target system.
- The executor must have access to the load files. The location of the files either can be specified in the load request or can be a default in the volatile data base.
- The target node must be able to recognize the trigger message or must be triggered manually.
- The circuit involved in the load operation must be enabled to perform service functions. It must also be in the ON or SERVICE state. For example, the following command readies circuit DMC-0 for down-line loading node BANGOR:

```
NCP>SET CIRCUIT DMC-0 SERVICE ENABLE STATE ON
```

4.1.3 NCP Down-line Load Commands

Use one of the following commands to initiate a down-line load:

NCP LOAD NODE
NCP LOAD VIA
NCP TRIGGER NODE
NCP TRIGGER VIA

The NCP LOAD and TRIGGER commands initiate down-line loads differently.

- Use the LOAD command when the target's primary bootstrap ROM is already running. The node that issues a LOAD command is always the executor for the requested down-line load (see Section 4.1).
- Use the TRIGGER command when the target is an unattended system equipped with an appropriate bootstrap ROM (see Section 4.1.2). The executor for a triggered down-line load is not necessarily the same node that issued the TRIGGER command. This command triggers the bootstrap mechanism of a node so that it loads itself.

Reflecting these differences, each command has a different parameter list. The LOAD command's parameter list includes all the parameters described in the next section. Any parameter you specify explicitly overrides the corresponding entry (if any) in the volatile data base. Conversely, every parameter you omit defaults to entries in the volatile data base.

In contrast, the NCP TRIGGER command's parameter list contains a subset of the parameters described below (node ID, circuit ID, physical address, and service password). To use this command or to trigger a load manually from the target, the executor's volatile data base must contain appropriate entries for the remaining parameters. Otherwise the executor does not have sufficient data to respond to the load request.

Note that the LOAD VIA and TRIGGER VIA commands are versions of LOAD and TRIGGER that allow you to load a target via a specified circuit. The target's identification is then obtained from the volatile data base. See Chapter 6 for descriptions of these commands and examples of their use.

4.1.4 Down-line Load Parameters

You can define default parameters for down-line loading in the permanent data base. Each target system to be loaded has a separate set of defaults. The defaults for a specific target apply whether the load is initiated by a command from the executor or manually from the target. Use the CFE DEFINE NODE command to set up the permanent data base, which is then loaded into the volatile data base when the system is loaded. To remove node parameters from the permanent data base, use the CFE PURGE command.

The down-line load parameters that you can define for each target in the permanent data base include:

- A service circuit parameter for the executor node

  SERVICE CIRCUIT *circuit-id*

- Parameters that pertain to the target node

  NODE *node-id*
  NAME *node-name*
  SERVICE NODE VERSION *phase*
  SERVICE DEVICE *device-type*
  SERVICE PASSWORD *password*
  HARDWARE ADDRESS *E-address* (Ethernet only)
  HOST *node-id*

  Note that the NODE, NAME, and HOST parameters used for the load are included in the message that the executor sends to the target when the load completes (see Section 4.1).

- Parameters that specify load files

  LOAD FILE *file*
  SECONDARY LOADER *file*
  TERTIARY LOADER *file*
  DIAGNOSTIC FILE *file* (Ethernet only)

All of these parameters are described in the following sections.

4.1.4.1 The Service Circuit Parameter — The following parameter specifies a circuit that links the executor node to the adjacent target node:

- SERVICE CIRCUIT is the circuit over which the load operation is to occur.

4.1.4.2 The Target Node Parameters — The following parameters relate to the target node:

- NODE identifies the target node by name or address. You can set up a separate down-line load data base entry in the permanent data base for each target node to be down-line loaded.

- NAME associates a node name with a target node address specified in the previous parameter.

- SERVICE NODE VERSION describes the target node as either Phase III or Phase IV. If you do not define this parameter either in the data base or in a command, it defaults to Phase IV.

- SERVICE DEVICE specifies the controller on the target node end of a service circuit. The service device handles down-line loading in a variety of ways, depending on the device used. The default secondary and tertiary loaders for the load operation depend on the designated service device. This parameter is therefore required for a down-line load if secondary and tertiary load files are not defined in the volatile data base. (See the description of secondary and tertiary load files in Section 4.1.4.3.) For example, the following command identifies the service device as a DMC device controller:

```
NCP>SET NODE BANGOR SERVICE DEVICE DMC
```

- SERVICE PASSWORD specifies a default service password, required for some devices to trigger the primary bootstrap mechanism on the target node. Depending on the service device, the password must be a hexadecimal number in one of two ranges. For DMC, DMR, DMP, or DMV devices, the range is 0 to FFFF. For UNA or QNA devices, the range is 0 to FFFFFFFFFFFFFFFF. For example, the following command sets the service password for a node with a load device in the first category:

```
NCP>SET NODE BANGOR SERVICE PASSWORD FEFE
```

- HARDWARE ADDRESS specifies the Ethernet hardware address originally assigned to the DEUNA or DEQNA controller for the target node. This address identifies an Ethernet node before it has started up and set a logical address for itself, called the physical address. (See Chapter 2 for a description of Ethernet addressing.) You can define the hardware address in either the permanent or the volatile data base, but not in a LOAD or TRIGGER command. The corresponding LOAD or TRIGGER command parameter is PHYSICAL ADDRESS.

  If you do not specify a PHYSICAL ADDRESS parameter in a down-line load command, the HARDWARE ADDRESS must be defined in the volatile data base. In this case, DECnet-RSX attempts to use that HARDWARE ADDRESS to communicate with the target node. If the attempt fails, DECnet-RSX tries again using an Ethernet address it has created from the target's node address.

- HOST specifies the node that will load task files down line to the target node. The host can be the executor node (the default) or any other reachable node other than the target node itself. At the end of the load sequence, the target receives a message with the name of the host and places that name in its volatile data base. The target can then use the host node for down-line task-loading applications. In addition, RSX tasks running on the target can issue connect requests to $HOST (see Section 4.3).

4.1.4.3 Down-line Load Files — The following parameters specify various load files:

- LOAD FILE specifies the file that contains the RSX-11S stand alone or DECnet-11S system image for loading down line to the target (see Section 4.1.1).
- SECONDARY LOADER and TERTIARY LOADER specify files containing programs that are part of the load sequence. If the volatile data base contains secondary and tertiary file name entries, they are the LOAD or TRIGGER command defaults for the SECONDARY and TERTIARY parameters. However, if the volatile data base does not contain entries for these parameters and if you omit them from a LOAD or TRIGGER command, the executor selects a secondary and a tertiary file according to the type of service device specified for the target. The executor chooses secondary and tertiary loader files from the list in Table 4-1. These files are also shipped with the DECnet-11S kit.
- DIAGNOSTIC FILE (Ethernet only) specifies a file containing diagnostics that the target server system can request.

Table 4–1: Default Loader Files by Target Device Type

Device Type	Secondary Loader	Tertiary Loader
DLV	SECDLV.SYS	TERDLV.SYS
DL11	SECDL.SYS	TERDL.SYS
DMC11	SECDMC.SYS	TERDMC.SYS
DMP	SECDMP.SYS	TERDMP.SYS
DMV	SECDMV.SYS	TERDMV.SYS
DPV	SECDPV.SYS	TERDPV.SYS
DU11	SECDU.SYS	TERDU.SYS
DUP11	SECDUP.SYS	TERDUP.SYS
DUV11	SECDUV.SYS	TERDUV.SYS
QNA	SECQNA.SYS	TERQNA.SYS
UNA	SECUNA.SYS	TERUNA.SYS

4.2 Up-line Dumping RSX-11S Memory

You can include certain SET NODE parameters in the permanent data base that allow an adjacent RSX-11S or server node to dump its memory into a file on the local DECnet-11M or DECnet-11M-PLUS node. (A target node can also dump its memory up line to an adjacent DECnet-VAX node. See the *DECnet-VAX System Manager's Guide* for details.) This procedure is referred to as up-line dumping. It is a valuable tool for crash analysis; that is, programmers can analyze the dump file and determine why the RSX-11S system failed. When it detects an impending system failure, the RSX-11S node will request the up-line dump if the appropriate support is selected during the RSX-11S system generation. The *DECnet-RSX Network Generation and Installation Guide* describes procedures for generating the appropriate RSX-11S support.

4.2.1 Up-line Dump Requirements

Before attempting an up-line dump operation, you must ensure that the nodes, lines, and circuits meet the following requirements:

- The target node must be directly connected to the executor node by a physical line. The executor node provides the line and circuit level access.
- The target node must be capable of requesting the up-line dump when it detects a system failure. If the dumping program (NETPAN) does not exist on the target, up-line dumping cannot occur.
- The executor node circuit involved in the dump operation must be enabled and set to ON to perform service functions. This is the default state of the circuit following NETGEN. For example, the following command readies circuit DMC-0 for up-line dumping:

  ```
  NCP>SET CIRCUIT DMC-0 SERVICE ENABLE STATE ON
  ```

- If the target does not supply the DUMP COUNT value (see Section 4.2.2), the executor must have this value in its volatile data base. RSX-11S systems built with the NETPAN routine always include the count.
- The executor must have a DUMP FILE entry in the volatile data base.

4.2.2 Up-line Dump Data Base Parameters

You can define the following up-line dump parameters in the permanent data base using the CFE DEFINE NODE command.

- DUMP FILE *file* specifies the file on the host node that will receive the RSX-11S memory dump.
- DUMP ADDRESS *address* specifies the octal address in the target's memory at which to begin the dump.
- DUMP COUNT *number* specifies a default decimal number of memory units to be dumped.

4.2.3 Up-line Dump Procedures

This section describes the procedures for an up-line dump initiated by a target node. DECnet uses the Maintenance Operation Protocol (MOP) to perform an up-line dump operation. MOP is a subset of the Digital Data Communications Message Protocol (DDCMP), which sends messages used for circuit testing, triggering, down-line loading, and up-line dumping. Refer to the *DNA Maintenance Operations Functional Specification* for a more complete discussion.

There are four steps involved in the up-line dump process. The actual dump takes place by repeating step 3, as described below:

1. When a target node senses a system failure, it sends a memory dump request to the executor node. The request is a MOP request dump service message. This message might contain information about the target's memory size (DUMP COUNT) and the up-line dump device type at the target.
2. If the message from the target includes a DUMP COUNT value, the executor uses it. Otherwise, the executor checks the target node's entry in its volatile data base for the DUMP COUNT, the target address from which to start dumping (DUMP ADDRESS), and the file where the memory will be stored (DUMP FILE) for the target. (If no entry exists for DUMP ADDRESS, the value defaults to 0.) The executor then sends a MOP request memory dump message to the target with the starting address and buffer size values.
3. Using the values it received from the executor, the target returns the requested block of memory in a MOP memory dump data message. The executor receives the block of dump data, places it in the DUMP FILE, increments the DUMP ADDRESS by the number of locations sent, and sends another request memory dump message to the target. This sequence is repeated until the amount of memory dumped matches the DUMP COUNT value.
4. Once the up-line dump is completed, the executor node automatically attempts to down-line load the target system. It initiates the down-line load by sending a TRIGGER message to the target.

4.3 Down-line Task Loading

Down-line task loading extends nonresident initial load, checkpointing, and overlay support to a DECnet-11S node. These functions are provided by the satellite loader (SLD) on the DECnet-11S node and by the the Host Task Loader (HLD) on the host DECnet node. The host node in this context is determined by the HOST parameter in effect when you down-line load the target (see Section 4.1.4.2). Neither HLD nor the remote SLD has an operator interface. Once the configuration is set up, they operate transparently. Figure 4-1 illustrates one instance of the down-line task-loading capability.

Section 4.3.1 discusses prerequisites for down-line task loading. Sections 4.3.2 and 4.3.3 discuss checkpointing and overlaying RSX-11S tasks.

4.3.1 Prerequisites for Down-line Loading RSX-11S Tasks

Down-line task loading requires communication between the SLD and HLD tasks. The SLD task is built during the RSX-11S network generation procedure. The HLD task and its mapping tables are built during the network generation procedure for a DECnet-RSX host or by VAX/VMS for a DECnet-VAX host (see Chapter 8).

The prerequisites for down-line task loading are:

- SLD must be installed and fixed into the RSX-11S system image using VMR; for example,

  ```
  >VMR
  Enter filename: RSX11S
  VMR>INS[xxx,64] SLD/FIX=YES
  VMR>LOA OV:
  ```

  This establishes SLD as the loading task (LDR...) for the RSX-11S executive. RSX-11S task image files are specified during network generation. These files are stored under the network UIC on the host system. The network UIC is [*xxx*,64]. OV: is the overlay pseudodevice.

- Any tasks to be down-line loaded into or checkpointed from the RSX-11S system must be installed, but not fixed, in the RSX-11S system image using VMR; for example,

  ```
  >VMR
  Enter filename: RSX11S
  VMR>INS [xxx,64]TLK/TASK=TLK
  ```

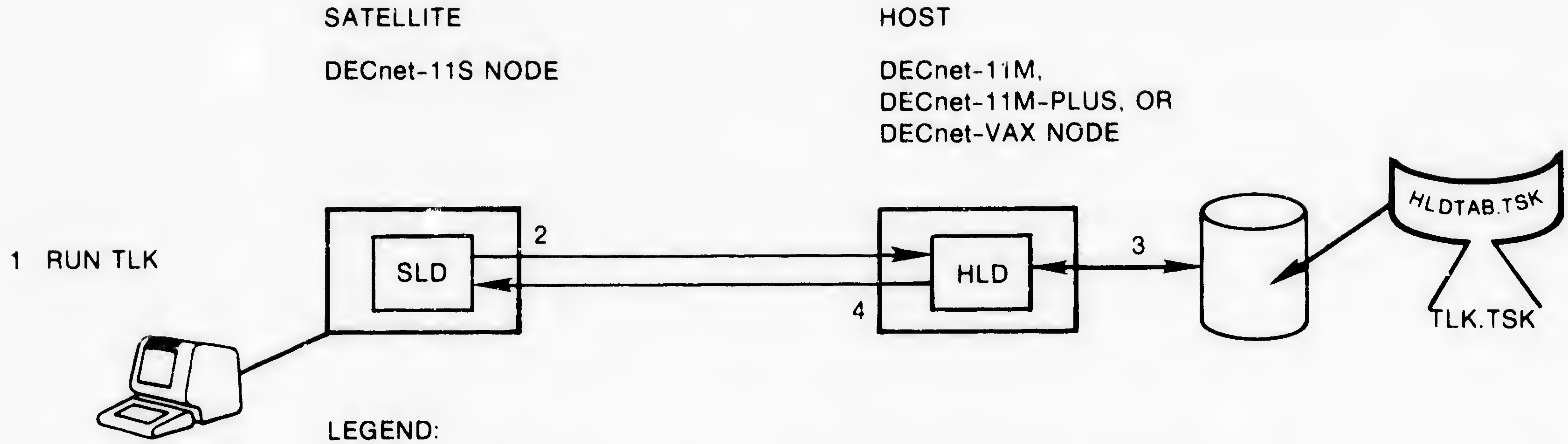
  In this example, typing RUN TLK on a terminal at the RSX-11S remote system initiates the down-line task load of the file [*xxx*,64]TLK.TSK.

- If the RSX-11S operating system is not to be down-line system loaded, you must define the node to which SLD is to connect, using the VNP SET EXECUTOR HOST command; for example,

  ```
  >VNP RSX11S
  VNP>SET EXECUTOR HOST BOSTON
  ```

  In this example, BOSTON is the name of the host node on which HLD resides.

- HLD, the Host Task Loader task, must be installed on the host node system.



LEGEND:

1 User types request.

2 SLD forwards request to HLD.

3 HLD retrieves TLK task.

4 HLD down-line loads TLK task.

Figure 4–1: Down-line Task Loading

4.3.2 Checkpointing RSX-11S Tasks

Checkpointing allows the interruption of an executing down-line loaded task when a higher priority task in the same partition becomes active. The interrupted task is written from the RSX-11S main memory to checkpoint space on the host. Then, the higher priority task is loaded into the partition and activated. When the higher priority task exits, the interrupted task is restored into main memory, where it continues to execute.

Checkpointing implies that a job is already running in the partition. Checkpoint space must be allocated inside the task image being down-line loaded (via the /AL switch during RSX-11S task build). Refer to the *RSX-11M/M-PLUS Task Builder Manual* for details.

4.3.3 Overlaying RSX-11S Tasks

Disk overlaying allows the execution of segments of a task to reduce the memory or address space needed to run on an RSX-11S system. HLD and SLD automatically handle the reading of overlay segments by satellite systems. You can specify this SLD capability during network generation for the RSX-11S system.

Chapter 5
Choosing Network Buffer Parameters

DECnet-RSX software uses several types of buffers to perform network communications. The buffers are used to store messages being sent and/or received between DECnet nodes. The permanent data base contains network parameters that determine the size, number, and memory allocation for these buffers. During network generation, either you or the NETGEN procedure itself determines initial values for these parameters. Subsequently, you can use CFE DEFINE SYSTEM commands to redefine them as necessary. The network buffer parameters include

- The size and maximum number of large buffers
- The maximum number of small buffers
- The maximum number of control buffers
- The minimum number of receive buffers

This chapter describes these parameters and provides guidelines for defining them so that your node will operate as efficiently as possible.

5.1 The NETCFE.CMD Command File

The NETCFE.CMD command file can be used as input to CFE. The file contains a series of CFE commands that defines the CETAB.MAC configuration as created by NETGEN. Included in the file are commands that define the network parameters discussed in this chapter. NETCFE.CMD can be used as a template for making changes to CETAB.MAC. In this case, revise a copy of the file and keep the original NETCFE.CMD for reference. Figure 5-1, an excerpt from a NETCFE.CMD file, shows CFE commands that define network buffers.

```
;
; CFE command file, define the generated CETAB configuration
;
CETAB
;
; General network configuration
;
                  .
                  .
                  .
                  .

DEFINE SYSTEM MAXIMUM CONTROL BUFFERS 6
DEFINE SYSTEM MAXIMUM LARGE BUFFERS 7
DEFINE SYSTEM MAXIMUM SMALL BUFFERS 11
DEFINE SYSTEM MINIMUM RECEIVE BUFFERS 2
DEFINE SYSTEM LARGE BUFFER SIZE 280
DEFINE EXECUTOR SEGMENT SIZE 262
                  .
                  .
                  .
                  .
```

Figure 5-1: Excerpt from a Sample NETCFE.CMD File

Chapter 6 describes the CFE DEFINE SYSTEM command, which you use to modify network parameters in the permanent data base.

5.2 Allocating Memory for Buffer Space

Deciding how much memory to dedicate to buffer space always involves trade-offs between minimizing memory usage and optimizing performance. For most users, the values allocated by NETGEN are adequate for establishing reasonable buffer resource requirements for support of the node being generated. The information provided here is for users who want to further optimize both memory utilization and DECnet performance.

Table 5-1 provides the following information about each buffer type:

- How the buffers are used
- Where the buffers are located
- The size of the buffers
- Factors that determine the number of buffers required

All DECnet buffers, with the exception of control buffers, are located in the network buffer pool. The default buffer pool is named POOL.. and is created as a common partition.

All DECnet processes must be mapped to control buffers. To facilitate this mapping, CCBs are allocated in executive address space.

Table 5-1: DECnet Buffers

	Large Buffers	Small Buffers	Control Buffers
Use	All receive messages, user data transmissions, logical link connects	User interrupt messages, ECL link service messages, node *talker* messages	DDCMP control messages, DECnet interprocess communication, one for each active SDB and LDB
Location	Network buffer pool, common partition POOL..	Common partition (POOL..) in network buffer pool	Executive address space (DSR)
Size (in decimal)	Variable (user specified, determines ECL segment size)	34 bytes	38 bytes
Factors Determining Number of Buffers Required	Number of lines, device types, number of logical links simultaneously active and performing data transfers, transmit traffic	Number of logical links simultaneously active and number of lines	Number of LDBs and SDBs

5.3 Large Buffers

DECnet-RSX software uses large buffers (LDBs) for intermediate storage of all user data being sent (transmit buffers), for storage of all incoming messages (receive buffers), and on routing nodes, for storage of all route-through packets. During NETGEN, the user specifies the size of LDBs. NETGEN itself calculates the number of LDBs to be allocated based on this and other NETGEN specifications.

5.3.1 Determining LDB Size

This section discusses factors in determining the optimum LDB size for your node. The LDB size that you define at NETGEN determines the node's segment size, which is the size into which large user messages are divided for end-to-end transmission. NETGEN calculates the segment size to be *ldbsize*-18. After NETGEN, use CFE or NCP to change segment size (DEFINE or SET EXECUTOR SEGMENT BUFFER SIZE).

If the segment sizes of communicating nodes — the nodes at either end of a logical link — differ, both nodes use the smaller of the two sizes. For communication to be successful, the chosen segment size must not be too large to be handled by intermediate nodes — that is, nodes that fall in the routing paths between source and destination. The LDB size for a routing node equals the maximum size of route-through packets that the node can handle. If all nodes within a network have the same LDB size, route-through packets too large to handle will not cause end-to-end communications to fail.

In some cases it may be necessary to reduce the local node's segment size to ensure that transmitted packets can successfully traverse the network. Likewise, it may be necessary to increase the local LDB size to ensure that your node can handle route-through packets.

Additional factors you must consider in defining LDB size include

- The record-blocking factors used by your application programs
- The amount of memory that you wish to allocate to LDBs
- The error rate of the communication lines
- Your throughput requirements

For example, if your applications are blocking data into 256-byte records and the LDB size is 256, DECnet software must split each record into two segments in order to transmit it. With an 18-byte ECL header affixed to each segment, one segment would have 238 bytes of data and an 18-byte header, and the other segment would have 18 bytes of data and an 18-byte header. In this type of situation, you would be incurring the overhead of segmenting records without using the segments efficiently, and at the same time you would be using up more memory than necessary for buffer space.

The obvious solution would be to increase the LDB size defined for the end nodes and thereby establish a larger segment size that could accommodate the 256-byte records without splitting them. However, this action would add to the amount of memory allocated to LDBs — something that you might wish to avoid. Also, as the size of the segments increases, the effective throughput rate (data received without error versus data received with errors) will decrease. Thus, to resolve the situation in a manner that would best suit your requirements, you might establish a LDB size (plus the ECL header size of 18 bytes) that is an even multiple of the record size used by your application programs. In this way you would minimize the amount of memory allocated to LDBs and increase the throughput.

5.3.2 Number of LDBs to Allocate

Major factors that determine the number of LDBs you should allocate include

- The number and type of local devices and controllers
- The amount of route-through traffic anticipated if yours is a routing node
- The number of logical links likely to be active simultaneously

The last row in Table 5-1 summarizes these factors, which you can use to help determine how many LDBs you need to maintain a reasonable level of performance. For most nodes, some trial and error is generally required to arrive at an optimum number.

Note that NETGEN itself calculates the minimum number of LDBs to be reserved for use as receive buffers (called the minimum receive threshold). When the number of available LDBs is equal to or less than the threshold, only requests for receive buffers will be serviced. In this way, receive traffic gets priority over transmit traffic. (Requests for transmit buffers will be queued until the number of available LDBs is above the threshold.) The receive buffer threshold should be set no higher than the sum of the buffers required for all lines on the system. To redefine the minimum receive threshold, use the CFE DEFINE SYSTEM MINIMUM RECEIVE BUFFERS command.

Table 5-2 gives the recommended number of LDBs for each communications device supported by DECnet-RSX.

Table 5-2: LDBs Required by Different Device Types

Device Type	Number of LDBs Used by Device
DL,DLV	1
DV	1 per line
DZ,DZV	1 per line
KDZ	1 per line
KDP	1 per line
DUP,DU,DUV	1
PCL	1
DMC (low speed, up to 19.2K bits/second)	2
DMC (high speed, 56K to 1M bits/second)	4
UNA	6
QNA	6

Routing nodes require more buffers than nonrouting nodes to provide space for route-through traffic. If there are insufficient LDBs, route-through packets will be discarded, resulting in degraded end-to-end performance. The following algorithm can be used to calculate the number of buffers required for a particular configuration to support route-through traffic. It assumes that route-through traffic contributes less than 60 percent of the total bandwidth of all lines and will give a low probability of packets being discarded.

number-of-buffers = 5 x SQRT(*n*)

where *n* is the number of lines.

Nonrouting nodes do not require buffers to support the route-through capability and can be configured with more large buffers than required for the device (see Table 5-2). In this case, the receive buffer threshold should be set to the value required for the device (as indicated in Table 5-2) or 2, whichever is lower.

5.4 Small Buffers

DECnet-RSX software uses small buffers (SDBs) to send user interrupt messages, End Communications layer (ECL) link service messages, and some control messages. SDBs have a fixed length and are used in place of LDBs in order to optimize resource utilization. NETGEN calculates the number of SDBs to be allocated.

At least one SDB is required for each of the logical links that are in use simultaneously. An additional SDB should be allocated for each physical link.

NETGEN calculates a default number. Then, when the system is in operation, you can use the Network Display Program (NTD), described in Chapter 11, to monitor the number actually in use. If necessary, you can adjust the buffer allocations using CFE.

5.5 Control Buffers

Each active LDB and SDB, while it is allocated, needs to have a control buffer (CCB) associated with it. DECnet-RSX software also uses CCBs to pass parameters between processes and to send DDCMP control messages. CCBs are fixed length. NETGEN calculates the minimum number of CCBs to be allocated. If more are needed, they are allocated dynamically.

One CCB is required for each active LDB and SDB. One CCB is generally required for each logical link. An additional CCB should also be allocated for each of the lines that will simultaneously be in the STARTING state (that is, set to the ON state).

NETGEN calculates this value. Use NTD to monitor the number of active logical links and to detect allocation failures. You have reached the optimum number when an allocation failure occurs only occasionally.

PART II
CFE, NCP, and VNP

This section describes the format, parameters, and usage of all CFE, NCP, and VNP commands. A summary of these commands is provided in Appendix A.

Chapter 6
Network Management Utilities (CFE, NCP, VNP)

This chapter describes the network management utilities and summarizes their commands. CFE commands modify the permanent data base, while NCP commands modify the volatile (running) data base. VNP commands are used to modify the system image file. Each utility is described below, including how to invoke it and how to exit it. Details on how to use the commands and individual command descriptions follow.

6.1 Configuration File Editor (CFE)

The Configuration File Editor (CFE) is a privileged utility program that accepts commands from a terminal or from an indirect command file to modify or display parameters in the configuration file, CETAB.MAC, which is produced during network generation. CFE commands operate exclusively on the permanent data base. Note that all CFE commands are privileged. To invoke CFE, you must be privileged to access the CETAB.MAC file.

6.1.1 Invoking CFE

Before invoking CFE, you must know the device and UIC under which CETAB.MAC resides. After network generation, CETAB.MAC resides under one of the following UICs:

DECnet-RSX-11M/M-PLUS	[*x*,54]
DECnet-RSX-11S	[*x*,64]

where *x* is the network UIC selected during network generation.

A privileged user can access CFE by using the RUN command:

```
MCR>RUN CFE
```

CFE prompts

```
Enter filename:
```

Enter the name of the configuration file you want to modify. If you omit any part of the file specification, the default is SY:[*current-uic*]CETAB.MAC. CFE then prompts

```
CFE>
```

Enter your CFE command after the prompt and press RETURN.

6.1.2 Exiting CFE

There are three ways to exit CFE:

1. Use the EXIT command or CTRL/Z in response to the CFE prompt to terminate an editing session and update the configuration file.

   ```
   CFE>EXIT
   MCR>
   ```

2. Use the EXIT PURGE command in response to the CFE prompt to terminate an editing session, update the configuration file, and delete the old file.

   ```
   CFE>EXIT PURGE
   MCR>
   ```

3. Use the KILL command in response to the CFE prompt to abort an editing session. The configuration file is not updated, and the old file is retained.

   ```
   CFE>KILL
   MCR>
   ```

6.2 Network Control Program (NCP)

The Network Control Program (NCP) is a utility program that accepts commands from a terminal or from an indirect command file to load, control, monitor, and test the network software, as well as to down-line load an RSX-11S system.

There are two versions of NCP: a full set of commands for DECnet-11M and M-PLUS systems and a subset of these for DECnet-11S. Both sets of commands are described in detail in this chapter and are summarized in Appendix A.

You can execute most NCP commands both locally and remotely (see Section 6.2.3). Appendix A lists all NCP commands supported by DECnet-RSX and flags those that are not executable remotely. It also highlights those commands that are specific to DECnet-RSX only.

6.2.1 Invoking NCP

There are three ways to invoke NCP:

1. `MCR>NCP`

   NCP then prompts

   `NCP>`

   Enter your NCP command after the prompt and press RETURN.

2. `MCR>NCP` *command*

   where *command* is any valid NCP command. With this method, you return to MCR after the command is executed.

3. `MCR>NCP @`*command file* `[/TR]`

   where *command file* is the name of an indirect command file containing NCP commands and /TR is an optional switch that allows you to trace the commands as they execute. You can nest command files.

6.2.2 Exiting NCP

Use either the EXIT command or CTRL/Z in response to the NCP prompt to exit NCP.

6.2.3 Executing NCP Remotely

Most NCP commands can be issued at your terminal for remote execution. This feature is useful in many ways; for example, you can use it to modify parameters or display information at a remote node, or you can down-line load a node that is not adjacent to your system. Remote execution can be accomplished in two ways:

- To execute a single NCP command at a remote node, issue it with the TELL prefix (see the TELL command description in this chapter). Appendix A summarizes all NCP commands and flags the few that cannot be executed with the TELL prefix.
- To execute a series of NCP commands at a remote node, use the SET EXECUTOR NODE command to temporarily set a specified remote node as the executor. Any subsequent commands that you issue are executed at that node until you return execution to your own node by issuing a CLEAR EXECUTOR NODE command. (Both commands are described in this chapter.) If you exit NCP, control automatically reverts to the local node.

When you issue commands to be executed at a remote node, you may have to specify access control information to access the node. You can append the access control information to the node name, or you can specify an alias node name that includes access control information. Chapter 2 provides more information on access control and the use of aliases.

The remote node interprets all commands according to its own network management software and uses any required default parameters from its own data base. Therefore, you must issue the command as if it were originating at the remote node. Consult the appropriate DECnet documentation if the node is other than DECnet-RSX.

6.3 Virtual Network Processor (VNP)

The Virtual Network Processor (VNP) is a privileged utility program. It accepts commands from a terminal or from an indirect command file to modify or display the DECnet-RSX software configuration as contained on an RSX-11M or RSX-11S system image file. You cannot use VNP on a DECnet-RSX-11M-PLUS disk image. All VNP commands are privileged and operate exclusively on the system image file. VNP does not support commands that cause displays to be written to a file.

Before using VNP, you must use the virtual monitor routine (VMR) to set the NETUIC in the RSX system image file. Use one of the following commands:

VMR>SET /NETUIC=[*xxx*,54]	(RSX-11M)
VMR>SET /NETUIC=[*xxx*,64]	(RSX-11S)

VNP uses the NETUIC when it searches for the network software on the LB: device. Therefore, it is imperative that the proper NETUIC be set in the system image file and that LB: be assigned.

Commands issued to VNP require that you have previously run VMR on the target system's image. For example, if you intend to use the network on RSX-11S, you must set the executor state to ON with VNP. NCP on RSX-11S does not support this command. See Chapter 4 for an example of this use of VMR and VNP.

6.3.1 Invoking VNP

A privileged user can access VNP by using the RUN command:

```
MCR>RUN $VNP
```

VNP prompts

```
Enter filename:
```

Enter the name of the operating system file you want to modify. If you do not specify the device and UIC, the SY: device and the current UIC are used. (There is no default file name.) VNP then prompts

```
VNP>
```

Enter your VNP command after the prompt and press RETURN.

6.3.2 Exiting VNP

Use either the EXIT command or CTRL/Z in response to the VNP prompt to exit VNP.

6.4 Using the Network Management Utility Commands

The following sections contain general information on command usage.

6.4.1 General Command Format

Most commands consist of three parts: the command verb, a component on which the command operates, and one or more parameters that further qualify the action to be taken on the component. Examples of the general command format follow:

Command Verbs	Components	Parameter Lists
CLEAR	EXECUTOR	[HOST] [RECEIVE PASSWORD] [TRANSMIT PASSWORD]
CLEAR	$\left\{\begin{array}{l}\text{LINE } \textit{line-id} \\ \text{KNOWN LINES}\end{array}\right\}$	$\left\{\begin{array}{l}\text{ALL} \\ \text{COUNTER TIMER}\end{array}\right\}$

For each command, you must supply the verb, one component option, and (usually) one or more of the parameters. Generally, the order in which you specify parameters does not matter. However, some commands have parameters that are mutually exclusive for the component. For example, in the CLEAR LINE command above, you must select either ALL or COUNTER TIMER (as indicated by the braces), whereas you can select one or more of the parameters for CLEAR EXECUTOR (as indicated by the brackets). On many of the SHOW and LIST commands, you do not have to specify any parameters because there is a default.

6.4.2 Issuing Commands

Enter the command keywords separated by spaces.

Continuing a command. To continue a long command to the next line, insert a hyphen (-) where you wish to break the line; then enter RET. You can continue the command following the prompt on the next line.

Example:

```
CFE>DEFINE LINE DZ-1-2 DUPLEX FULL SPEED 9600 - RET
CFE>VECTOR 360 CSR 160120 RET
CFE>
```

Any necessary spaces in the command must be entered even if one occurs just before the hyphen.

Abbreviating command keywords. When entering a command, you can abbreviate any keyword to the first three or more unique characters. For example, the following two versions of the same command are equally valid:

```
NCP>SHOW MODULE X25-ACCESS ALL DESTINATIONS CHARACTERISTICS

NCP>SHO MOD X25-A ALL DES CHA
```

Note that X25-ACCESS is abbreviated to the first five unique characters to differentiate it from other entities beginning with the X25- character string. For clarity and consistency, all examples in this manual show the full command syntax.

Entering comment lines. You can enter comment lines by prefacing each line of text with a semicolon (;).

6.4.3 Using the Help Facility

Use the HELP command whenever you need assistance in selecting network management commands and parameter options. Execution of this command by any of the three utilities causes a summary of help text to be displayed.

Example:

```
NCP
NCP HELP
```

Use the HELP keyword with a selected command (and component) to display the portion of the help file relevant to that command.

Example:

```
>NCP
NCP>HELP LINE
```

6.4.4 Command Prompting

Command prompting provides additional assistance when using network management commands. At any point where you can type a space, type (RET) instead. This produces a prompt requesting the required information. If you need additional information after a prompt, you can type (ESC) again to display a shorter form of the help message.

Example:

```
NCP>SET (RET)
Set component: (ESC)
```

6.4.5 Exit Status and Error Handling

When CFE, NCP, or VNP terminates, it sets a bit that can be inspected using the normal RSX exit status handling techniques described in RSX-11M/M-PLUS operating system documentation. The utility sets the exit status to indicate the worst case for the commands executed. The possible statuses and their meanings are summarized below.

Status	Numeric Value	Meaning
Success	1	The command executed as requested.
Warning	0	Minor error. Processing continues.
Error	2	The command was accepted, but failed to execute completely. The system waits for a new command.
Severe Error	4	The command was not accepted. Processing stops and control returns to the monitor.

If the command executes successfully, you receive the utility prompt on the next line, whereupon you can issue another command. If the command is unsuccessful, you receive an error message on the next line indicating the reason for the error, followed by the utility prompt.

Example:

```
MCR>NCP
NCP>SET LINE DMC-0 TOP
NCP>LOAD NODE XYZ
NCP - Load failed, line communication error
NCP>
```

Error messages for all three utilities are provided in Chapter 14.

NOTE

The following NCP commands do not affect the exit status:

CLEAR EXECUTOR NODE
EXIT (or CTRL Z)
HELP
SET EXECUTOR NODE

6.5 Command Descriptions

The rest of this chapter is devoted to descriptions of the individual commands and their parameters. If you need more information on the command components or their parameters, refer to the tutorial chapters for a more in-depth discussion.

Commands are presented in alphabetical order. The command name and the utility (or utilities) to which it applies are printed at the top of the page (many NCP and VNP commands have the same format and are described together). The full DECnet-RSX-11M/M-PLUS set is described first. The RSX-11S subset is described at the end of the chapter (NCP/S appears at the top of those pages). All commands are designated as privileged (P) or nonprivileged (NP) at the top of the page next to the command name. On combined NCP/VNP commands, an NP designator applies to NCP commands only, as all VNP commands are privileged.

6.5.1 Graphic Conventions

Review the graphic conventions outlined at the front of this manual, especially the usage of braces { } and brackets [ ]. These conventions are used throughout the command descriptions to specify parameter selection and optionality. **Shaded text** indicates commands or parameters that are valid for PSI users only.

6.5.2 Standard Definitions

The following parameters and user-supplied variables are frequently used in the command descriptions. They are described here to avoid needless repetition. Exceptions to these definitions are called out in individual parameter descriptions.

acc-con-info
: Access control information in one of the following forms:

  /user-id[*/password*[*/account*]]

  USER *user-id* [PASSWORD *password* [ACCOUNT *account*]]

  NOTE: DECnet-RSX does not use the account field. For more information on access control, see Chapter 2.

circuit-id
: A DECnet circuit in the form *dev-c*[*-u*][*.t*]. For more information on circuit identification, see Chapter 2.

csr-address
: CSR address in the range of 160,000 to 177,776 (octal).

dest-name
: 1 to 6 alphanumeric characters.

dte-address
: 1- to 15-digit hexadecimal integer address.

E-address
: A string of 12 hexadecimal digits in the form *nn-nn-nn-nn-nn-nn*.

E-node-name
: See description for *node-name*, below.

file
: A file specification, which is interpreted according to the executor node's file system. The standard format for RSX is *dev*:[*uic*]*filename.type;ver*.

group-name
: 1 to 6 alphanumeric characters.

line-id
: A line in the form *dev-c*[*-u*]. For more information on line identification, see Chapter 2.

milliseconds
: A decimal integer in the range of 1 to 65,535.

node-address
: A decimal integer in the range of 1 to 1023

node-id
: A node name (see *node-name*) or node address (see *node-address*). For more information on node identification, see Chapter 2.

node-name
: 1 to 6 alphanumeric characters, including at least 1 alphabetic character.

password	A hexadecimal number. For DDCMP circuits, it is in the range of 0 to FFFF; for Ethernet circuits, it is in the range of 0 to FFFFFFFFFFFFFFFF (up to 16 hexadecimal digits).
range	One or two 1- to 4-digit integers in the range of 0 to 9999. If you specify two subaddresses, they must be separated by a hyphen (-), and the second must be greater than the first (e.g., 10-100). See *PSI Network-specific Information* for the maximum length of a subaddress on your network.
seconds	A decimal integer in the range of 1 to 65,535.
CHARACTERISTICS	Displays static information for the specified component. This type of information is automatically displayed for all CFE LIST commands.
COUNTERS	(Invalid for VNP) Displays error and performance statistics for the specified component. See Appendix E for a summary of network counters.
STATUS	(Invalid for VNP) Displays dynamic information for the specified component.
SUMMARY	(Default) Displays only the most useful information for the specified component.
SCOPE *scope*	(Valid for privileged users only) Qualifies the extent to which the specified operation applies (default: requestor's terminal). Possible values for *scope* are: GLOBAL — The operation applies to all terminals. TERMINAL *term-id* — The operation applies to the specified terminal only, where *term-id* is a standard RSX terminal device identification.
TO *file*	(Valid for NCP only) Directs output to the specified file or device at the local node (default: information displayed at the calling terminal). If you specify an existing file as output, the information is appended to that file. Information is transferred in ASCII in the same format as it is displayed at the terminal. (See the description of *file*, above, for format.)

6.5.3 RSX-11M/M-PLUS CFE, NCP, and VNP Command Summary

This section describes all of the CFE and NCP commands supported by RSX-11M/M-PLUS. VNP commands supported by RSX-11M are described in tandem with the corresponding NCP command. RSX-11S NCP commands are described in Section 6.5.4.

Use the NCP CLEAR ALIAS command to remove the specified alias node name(s) from the volatile data base. Use the VNP command to do the same with the system image file.

Format:

```
CLEAR { ALL ALIASES         }  [[SCOPE] scope]
      { ALIAS alias-name    }
      { KNOWN ALIASES       }
```

where

ALL ALIASES	(Valid for privileged users only) Removes all alias names.
ALIAS *alias-name*	Removes only the specified alias name.
KNOWN ALIASES	Removes all alias names for the specified scope.
SCOPE *scope*	(Invalid for ALL ALIASES) See definition in Section 6.5.2.

Example:

```
NCP>CLEAR ALIAS NY5 SCOPE TERMINAL TT2:
```

This command removes the association of alias NY5 with terminal TT2:.

Use the CLEAR CIRCUIT command to zero the specified PSI circuit counter timer(s) in the volatile data base.

Format:

CLEAR {CIRCUIT *circuit-id* | KNOWN CIRCUITS} COUNTER TIMER

where

CIRCUIT *circuit-id*	Zeroes the counter timer for the specified circuit only.
KNOWN CIRCUITS	Zeroes the counter timers for all known circuits.

Example:

```
NCP>CLEAR CIRCUIT PVC1 COUNTER TIMER
```

This command zeroes the counter timer for PSI circuit PVC1.

Use the NCP CLEAR EXECUTOR command to remove specified executor node parameters from the volatile data base. Use the VNP command to do the same with the system image file.

Format:

```
CLEAR EXECUTOR   [HOST]
                 [RECEIVE PASSWORD]
                 [TRANSMIT PASSWORD]
```

where

HOST	Removes identification of the host for the executor node. The executor then defaults to itself as host for service operations.
RECEIVE PASSWORD	Removes the executor node's receive password.
TRANSMIT PASSWORD	Removes the executor node's transmit password.

Example:

```
NCP>CLEAR EXECUTOR RECEIVE PASSWORD
```

This command removes the receive password.

Use the CLEAR EXECUTOR NODE command to return NCP command execution to the local node after issuing an NCP SET EXECUTOR NODE command.

Restriction:

You cannot use the TELL prefix with this command.

Format:

CLEAR EXECUTOR NODE

Example:

```
NCP>SET EXECUTOR NODE BOS


NCP>CLEAR EXECUTOR NODE
```

The first command causes subsequent NCP commands to be executed at node BOS. The second command returns NCP command execution to the local node.

Use the NCP CLEAR LINE command to unload the specified line(s) and to remove all associated parameters from the volatile data base. Use the VNP command to do the same with the system image file. You can also use the NCP command to zero the counter timer(s) for the specified PSI line(s).

Restriction:

Before you unload a DECnet line, all circuits associated with the line must be in the OFF state (see SET CIRCUIT). Any PSI line to be unloaded must also be in the OFF state (see SET LINE and SET KNOWN LINES).

Format:

$$\text{CLEAR} \left\{ \begin{array}{l} \text{LINE } \textit{line-id} \\ \text{KNOWN LINES} \end{array} \right\} \left\{ \begin{array}{l} \text{ALL} \\ \textbf{COUNTER TIMER} \end{array} \right\}$$

where

LINE *line-id*	Clears the specified line only.
KNOWN LINES	Clears all known lines.
ALL	Unloads the specified line(s) and removes all associated line parameters.
COUNTER TIMER	(Invalid for VNP) Zeroes the counter timer(s) for the specified PSI line(s).

Examples:

```
NCP>CLEAR LINE DMC-0 ALL
```

This command unloads DECnet line DMC-0 and removes all parameters associated with it, thus making DMC-0 unusable.

```
NCP>CLEAR LINE SDP-1 COUNTER TIMER
```

This command zeroes the counter timer for PSI line SDP-1.

CLEAR LOGGING

Use the NCP CLEAR LOGGING command to remove logging parameters from the volatile data base. Use the VNP command to do the same with the system image file.

Restriction:

Whenever a circuit, line, module, node, or sink is specified in a CLEAR LOGGING command, an event list or an ALL EVENTS or KNOWN EVENTS parameter must also be included.

Format:

```
CLEAR {KNOWN LOGGING   } [NAME]
      {LOGGING CONSOLE }
      {LOGGING FILE    } [[ALL EVENTS    [{CIRCUIT circuit-id                }]]]
      {LOGGING MONITOR }  [EVENTS list   [{LINE line-id                      }]]]
                         {[KNOWN EVENTS  [{MODULE {X25-PROTOCOL}             }]]}
                          [              [{       {X25-SERVER  }             }]]]
                          [              [{       {X29-SERVER  }             }]]]
                          [              [{NODE node-id                      }]]]
                          [              [ SINK {EXECUTOR                 }  ]]]
                          [              [      {NODE {node-id}           }  ]]]
                          [              [      {     {$HOST  }           }  ]]]
```

where

KNOWN LOGGING	Removes parameters for all known logging components.
LOGGING CONSOLE	Removes parameters for logging to the console.
LOGGING FILE	Removes parameters for logging to a file.
LOGGING MONITOR	Removes parameters for logging to a monitor program.
NAME	Returns the logging component to its default (console: CO0:; file: LB:[1,6]EVENTLOG.SYS; monitor: MON...).
ALL EVENTS	Removes parameters for all logging event classes and types.
EVENTS *list*	Removes parameters for the logging event class and type(s) specified in *list*. See Chapter 2 for the event list format.
KNOWN EVENTS	Removes parameters for all logging events that DECnet-RSX can generate.
CIRCUIT *circuit-id*	(Invalid for VNP) Inhibits event logging for the specified circuit (see restriction).

LINE *line-id*	Inhibits event logging for the specified line (see restriction).
MODULE	(Invalid for VNP) Inhibits event logging for the specified module (see restriction).
NODE *node-id*	Inhibits event logging for the specified node (see restriction).
SINK	Inhibits logging of the specified event(s) at the specified node (see restriction).

EXECUTOR	(Default) Events are not to be logged at the executor node
NODE *node-id*	Events are not to be logged at the specified node.
NODE $HOST	(Invalid for NCP) Events are not to be logged at the host node.

Examples:

```
NCP>CLEAR LOGGING CONSOLE EVENT 2.1
```

This command ceases logging of event 2.1 to the console.

```
NCP>SET LOGGING CONSOLE EVENT 2.1 SINK NODE PHL
NCP>SET LOGGING CONSOLE NAME TT1: EVENT 2.1 STATE ON
                    .
                    .
                    .
NCP>CLEAR LOGGING CONSOLE EVENT 2.1 SINK NODE PHL
```

The first command causes 2.1 events to be logged on the console on node PHL. The second command enables logging of 2.1 events to TT1: on the local node. The third command reverts event logging to the local node only, and 2.1 events are no longer logged to the console on node PHL.

Use the CLEAR MODULE X25-ACCESS command to remove destination names from the volatile data base.

Format:

CLEAR MODULE X25-ACCESS { **ALL DESTINATIONS** / **DESTINATION** ***dest-name*** / **KNOWN DESTINATIONS** } **[[SCOPE]** ***scope*****]**

where

ALL DESTINATIONS	Removes all destination names, regardless of scope.
DESTINATION *dest-name*	Removes only the specified destination name.
KNOWN DESTINATIONS	Removes all destination names for the specified scope.
SCOPE *scope*	See definition in Section 6.5.2.

Examples:

```
NCP CLEAR MODULE X25-ACCESS ALL DESTINATIONS
```

This command removes all destination names.

```
NCP CLEAR MODULE X25-ACCESS DESTINATION CUG111 TERMINAL TT2:
```

This command removes destination name CUG111 associated with terminal TT2:.

Use the CLEAR MODULE X25-PROTOCOL command to zero the counter timer(s) for the specified DTE(s) or to remove from the volatile data base all parameters for the specified DTE(s) or group(s).

Format:

```
CLEAR MODULE X25-PROTOCOL { {DTE dte-address}    {ALL          } }
                          { {KNOWN DTES     }    {COUNTER TIMER} }
                          {                                      }
                          { {GROUP group-name}   ALL             }
                          { {KNOWN GROUPS    }                   }
```

where

DTE *dte-address*	Clears the specified DTE only.
KNOWN DTES	Clears all known DTEs.
GROUP *group-name*	Removes parameters for the specified group only.
KNOWN GROUPS	Removes parameters for all known groups.
ALL	Removes all parameters for the specified DTE(s) or group(s).
COUNTER TIMER	Zeroes the counter timer(s) for the specified DTE(s).

Examples:

```
NCP>CLEAR MODULE X25-PROTOCOL DTE 123456789 COUNTER TIMER
```

This command zeroes the counter timer for local DTE 123456789.

```
NCP>CLEAR MODULE X25-PROTOCOL KNOWN GROUPS ALL
```

This command removes parameters for all known groups.

CLEAR MODULE X25–SERVER CLEAR MODULE X29–SERVER

Use the CLEAR MODULE X25/X29-SERVER commands to zero the counter timer or to remove the specified destination name(s) and all associated parameters from the volatile data base.

Format:

CLEAR MODULE $\left\{\begin{array}{l}\textbf{X25–SERVER}\\ \textbf{X29–SERVER}\end{array}\right\}$ $\left\{\begin{array}{l}\textbf{COUNTER TIMER}\\ \left\{\begin{array}{l}\textbf{DESTINATION } \textbf{\textit{dest-name}}\\ \textbf{KNOWN DESTINATIONS}\end{array}\right\}\end{array}\right\}$

where

COUNTER TIMER	Zeroes the server module counter timer.
DESTINATION *dest-name*	Removes parameters for the specified destination only
KNOWN DESTINATIONS	Removes parameters for all known destinations.

Example:

```
NCP>CLEAR MODULE X25-SERVER DESTINATION CHI205 ALL
```

This command removes all parameters for destination CHI205.

Use the NCP CLEAR NODE command to remove the specified node parameter from the volatile data base. Use the VNP command to do the same with the system image file.

Format:

CLEAR NODE *node-id* $\left\{\begin{array}{l}\text{CIRCUIT}\\\text{NAME}\end{array}\right\}$

where

NODE *node-id*	Specifies the node whose parameters are to be removed. You must use a node name if you are removing the circuit identification.
CIRCUIT	Removes the circuit identification associated with the node name for node level loopback testing.
NAME	Removes the node name associated with the node identification.

Example:

```
NCP>CLEAR NODE TESTER CIRCUIT
```

This command removes the circuit identification associated with node name TESTER.

Use the NCP CLEAR OBJECT command to remove all object parameters from the volatile data base. Use the VNP command to do the same with the system image file.

Format:

CLEAR {OBJECT *type-code* | KNOWN OBJECTS} ALL

where

OBJECT *type-code*	Removes parameters for the specified object only. Use an object type code from Appendix B.
KNOWN OBJECTS	Removes parameters for all known objects.
ALL	(Required for NCP; optional for VNP) Specifies that all parameters are to be removed.

Example:

```
NCP>CLEAR OBJECT 245 ALL
```

This command removes all parameters for object 245.

Use the NCP CLEAR PROCESS command to unload a process and remove all process parameters from the volatile data base. Use the VNP command to do the same with the system image file.

Restrictions:

- You can unload a process only when it is not being used.
- If you clear NW from the volatile data base and then reload it using NCP SET PROCESS, you must also issue the SET MODULE X25-PROTOCOL command (first with STATE OFF and then with STATE ON) to update the process pointers to the data base.

Format:

CLEAR PROCESS *process-name*

where

PROCESS *process-name*	Specifies the process to unload. Use a process name from the table in Chapter 2.

Example:

```
NCP>CLEAR PROCESS DL
```

This command unloads the DLX process and removes all parameters associated with it.

Use the NCP CLEAR SYSTEM command to unload the Communications Executive, release any resources being used by the network, and remove all system parameters from the volatile data base. Use the VNP command to unload the Communications Executive in the system image file.

Restrictions:

- You cannot use the TELL prefix with this command.
- Before you issue NCP CLEAR SYSTEM, the executor must be in the OFF state (see NCP SET EXECUTOR).
- If you are running PSI, the X.25 server module must be in the OFF state (see SET MODULE X25-SERVER).

Format:

CLEAR SYSTEM

Use the CLEAR TRACE command to stop tracing of PSI frames on the specified line(s). For more information on the trace interpreter task, see Chapter 13.

Format:

CLEAR TRACE $\left\{\begin{array}{l}\text{LINE } \textit{line-id} \\ \text{ACTIVE LINES}\end{array}\right\}$

where

LINE *line-id*	Disables tracing on the specified line only.
ACTIVE LINES	Disables tracing on all active lines.

Example:

```
NCP>CLEAR TRACE LINE SDP-0
```

This command disables tracing of PSI frames on line SDP-0.

Use the DEFINE CIRCUIT command to modify circuit parameters in the permanent data base. You can also use this command to create parameters for PVCs.

Restriction:

The values used for MAXIMUM DATA and MAXIMUM WINDOW must be the same as the packet size and window size agreed upon by the network authorities for the PVC.

Format:

DEFINE {CIRCUIT *circuit-id* | KNOWN CIRCUITS}
[CHANNEL *channel-number*]
[COST *cost*]
[COUNTER TIMER *seconds*]
[DTE *dte-address*]
[HELLO TIMER *seconds*]
[MAXIMUM BROADCAST ROUTERS *number*]
[MAXIMUM DATA *byte-count*]
[MAXIMUM RECALLS *retry-count*]
[MAXIMUM WINDOW *block-count*]
[MULTIPOINT ACTIVE *active-ratio*]
[NUMBER *dte-address*]
[RECALL TIMER *seconds*]
[ROUTER PRIORITY *priority*]
[SERVICE {DISABLE | ENABLE}]
[STATE {OFF | ON}]
[TRIBUTARY *trib-address*]
[USAGE {INCOMING | OUTGOING}]

where

CIRCUIT *circuit-id*	Modifies (or creates) parameters for the specified circuit only.
KNOWN CIRCUITS	Modifies (or creates) parameters for all known circuits.
CHANNEL *channel-number*	Specifies the logical channel number for a PSI PVC (range: 0 to 4095).
COST *cost*	Specifies the routing cost of the circuit (range: 1 to 25). Messages will travel between nodes along the path with the smallest total cost.

COUNTER TIMER *seconds*	Sets a timer whose expiration causes a PSI circuit counter logging event.
DTE *dte-address*	Specifies a local DTE for a PSI PVC.
HELLO TIMER *seconds*	Specifies the frequency of routing hello messages sent to an adjacent node on the circuit.
MAXIMUM BROADCAST ROUTERS *number*	Specifies the maximum number of routing nodes that an Ethernet circuit can maintain (range: 0 to 32; default: 10).
MAXIMUM DATA *byte-count*	Specifies the maximum packet size for a PSI PVC. For DLM SVCs, this parameter specifies the maximum packet size to be negotiated with the network, if allowed. This parameter value (range: 16 to 1024) must be at least 5 bytes less than the maximum data size specified for PSI lines (see DEFINE LINE). By default, packet size takes the network value specified by the MAXIMUM DATA parameter in the DEFINE MODULE X25-PROTOCOL command. See *PSI Network-specific Information* for the network value of this parameter.
MAXIMUM RECALLS *retry-count*	Specifies the maximum number of times that DECnet should attempt to place a call to establish an SVC (range: 0 to 255; default: 5).
MAXIMUM WINDOW *block-count*	Specifies the maximum number of blocks allowed to be outstanding on the circuit for PSI PVCs. For DLM SVCs, this specifies the maximum number of outstanding blocks to be negotiated with the network, if allowed (range: 1 to 7). By default, window size takes the network value specified by the MAXIMUM WINDOW parameter in the DEFINE MODULE X25-PROTOCOL command. See *RSX-11 PSI Network-specific Information* for the network value of this parameter.
MULTIPOINT ACTIVE *active-ratio*	(Valid only for circuits with multipoint software) Specifies the rate at which a tributary in the active multipoint state is polled (range: 1 to 255).
NUMBER *dte-address*	(Valid for outgoing SVCs only) Specifies a remote DTE address used by DECnet when trying to establish an SVC.
RECALL TIMER *seconds*	Sets the interval to wait after a failed call before attempting to place another call to establish an SVC (default: 30).

ROUTER PRIORITY *priority*	Specifies the priority that the executor node on this circuit is to have in selecting a designated router for the circuit (range: 0 to 255).
SERVICE	Specifies whether the circuit is enabled or disabled for down-line loading and loopback testing.
STATE	Sets the circuit's operational state to OFF or ON. For circuits owned by the executor, the state of the corresponding line(s) must be ON.
TRIBUTARY *trib-address*	Specifies the data link physical tributary address of the multipoint circuit (range: 1 to 255).
USAGE	Specifies the availability of a DLM SVC for either incoming or outgoing switched calls.

Examples:

```
CFE>DEFINE CIRCUIT DZ-3 STATE OFF
```

This command makes circuit DZ-3 unavailable for use.

```
CFE>DEFINE KNOWN CIRCUITS MAXIMUM RECALLS 10
```

This command sets to 10 the maximum number of times DECnet should attempt to place a call on all known circuits.

DEFINE EXECUTOR

Use the DEFINE EXECUTOR command to modify executor node parameters in the permanent data base.

Format:

```
DEFINE EXECUTOR  [ADDRESS node-address]
                 [BROADCAST ROUTING TIMER seconds]
                 [HOST node-address]
                 [IDENTIFICATION id-string]
                 [MAXIMUM ADDRESS node-address]
                 [MAXIMUM BROADCAST NONROUTERS number]
                 [MAXIMUM COST number]
                 [MAXIMUM HOPS number]
                 [MAXIMUM LINKS number]
                 [MAXIMUM NODE COUNTERS number]
                 [NAME node-name]
                 [ROUTING TIMER seconds]
                 [SEGMENT BUFFER SIZE number]
                 [SUBADDRESSES range]
                 [VERIFICATION [STATE] {OFF}]
                 [                     {ON }]
```

where

ADDRESS *node-address*	Specifies the address of the executor node.
BROADCAST ROUTING TIMER *seconds*	Sets a timer whose expiration forces a routing update on Ethernet circuits, which produces a routing configuration message for each adjacent node.
HOST *node-address*	Specifies the host node address for down-line load, trigger, and up-line dump operations.
IDENTIFICATION *id-string*	Specifies the text identification string for the executor node (range: 1 to 32 characters). You must use quotation marks (") to delimit any string containing blanks or tabs. To indicate a quotation mark within a quoted string, use a double quotation mark (to distinguish it from a string delimiter).
MAXIMUM ADDRESS *node-address*	Specifies the maximum node address of the executor or any remote node known to the executor.
MAXIMUM BROADCAST NONROUTERS *number*	Specifies the maximum number of nonrouting nodes that the executor node can have on its Ethernet circuits (range: 0 to 65535; default: 64).

MAXIMUM COST *number*	Specifies the maximum total path cost allowed to any node (range: 1 to 1022).
MAXIMUM HOPS *number*	Specifies the maximum number of routing hops allowed from the executor node to any other node (range: 1 to 30).
MAXIMUM LINKS *number*	Specifies the maximum number of active logical links for the executor node (range: 1 to 255).
MAXIMUM NODE COUNTERS *number*	Specifies the maximum number of node counters allowed. If you require more counters, the existing ones are reassigned on demand in the same order as they were originally assigned. Use a decimal integer greater than 1, but not exceeding the executor maximum address (range: 1 to 170).
NAME *node-name*	Specifies the node name of the executor.
ROUTING TIMER *seconds*	Sets a timer whose expiration forces a routing update on non-Ethernet circuits.
SEGMENT BUFFER SIZE *number*	Specifies the maximum size of transmit buffers (in bytes), thereby controlling the size of the NSP message segment to be sent. This value is the maximum size message that the End Communications layer can transmit; it does not include routing or data link overhead (range: 1 to 4096; default: 18 bytes less than the large buffer size specified in the DEFINE SYSTEM command). The segment buffer size can never be greater than the large buffer size minus 18 bytes.
SUBADDRESSES *range*	Specifies a range of local DTE subaddresses that are valid on any PSI circuit for incoming calls to the executor node.
VERIFICATION STATE	Sets the logical link access control verification state for the executor node. OFF — The executor node does not verify access control on any incoming requests. ON — The executor node verifies access control on all incoming connect requests according to the options set for each object (see the discussion of access control in Chapter 2).

Examples:

```
CFE>DEFINE EXECUTOR ADDRESS 21
```

This command sets the executor address to 21.

```
CFE>DEFINE EXECUTOR MAXIMUM LINKS 20
```

This command sets the maximum number of active logical links to 20.

Use the DEFINE LINE command to modify line parameters in the permanent data base.

Format:

```
DEFINE {LINE line-id    } [CONTROLLER] CSR csr-address]
       {KNOWN LINES     } [CONTROLLER {LOOPBACK}]
                          [           {NORMAL  }]
                          [COUNTER TIMER seconds]
                          [DUPLEX {FULL}]
                          [       {HALF}]
                          [HOLDBACK TIMER milliseconds]
                          [MAXIMUM DATA byte-count]
                          [MAXIMUM RETRANSMITS retry-count]
                          [MAXIMUM WINDOW block-count]
                          [MULTIPOINT DEAD dead-ratio]
                          [PRIORITY hardware-priority]
                          [RETRANSMIT TIMER milliseconds]
                          [SPEED baud-rate]
                          [STATE {CLEARED}]
                          [      {OFF    }]
                          [      {ON     }]
                          [UNIT CSR csr-address]
                          [VECTOR vector-address]
```

where

LINE *line-id*	Modifies parameters for the specified line only.
KNOWN LINES	Indicates that parameters for all known lines are to be modified
CONTROLLER CSR *csr-address*	Specifies the address of the first CSR for the line controller.
CONTROLLER	(For Ethernet, DMC/DMR, DMP, and DMV lines) Specifies the controller mode for the line. LOOPBACK — Internal device loopback mode. NORMAL — Normal operating mode.
COUNTER TIMER *seconds*	Sets a timer whose expiration causes a PSI line counter logging event.
DUPLEX	(Not applicable for PSI or Ethernet lines) Sets the hardware transmission mode to full duplex or half duplex.

HOLDBACK TIMER *milliseconds*
Specifies the length of time that a PSI message acknowledgment can be held back in order to be included with another data message.

MAXIMUM DATA *byte-count*
Specifies the maximum frame size (in bytes) for the line(s). This value must be at least 5 bytes larger than the maximum data size specified for PSI circuits (see the DEFINE MODULE X25-PROTOCOL command) (range: 16 to 1024).

MAXIMUM RETRANSMITS *retry-count*
Specifies the maximum number of retransmissions of a frame over the specified PSI line(s) (range: 0 to 255).

MAXIMUM WINDOW *block-count*
Specifies the maximum number of frames for which outstanding PSI acknowledgment messages are allowed (range: 1 to 7).

MULTIPOINT DEAD *dead-ratio*
(Valid only for lines with multipoint software) Specifies the polling rate for a tributary in the dead multipoint state (range: 1 to 255).

PRIORITY *hardware-priority*
Specifies the hardware priority of the controller (range: 4 to 7).

RETRANSMIT TIMER *milliseconds*
Specifies the time to elapse before a block is retransmitted (range: 100 to 10,000). CFE rounds off all values within this range to the nearest hundred milliseconds.

SPEED *baud-rate*
Sets the line baud rate to one of the following:

50	300	2400
75	600	3600
110	1200	4800
134	1800	7200
150	2000	9600

To set a high speed device (that is, over 9600 baud), enter >9600 or the actual baud rate.

STATE
Sets the line's operational state when the system is loaded.

- CLEARED — The line is not loaded. A DDCMP controller line may not be set to CLEARED if any circuits on it are in the ON state.
- OFF — The line is loaded, but not available for use. A DDCMP controller line may not be set to OFF if any circuits on it are in the ON state.
- ON — The line is loaded and available for normal use except during temporary overrides for service operations. All PSI lines are set to ON. If the line is a DDCMP point-to-point line or a DDCMP tributary, the corresponding circuit is also set to ON.

UNIT CSR *csr-address*
: Specifies the address of the first CSR for a DECnet device controlled by a KMC-11.

NOTE

With the KDP, a unit CSR can be specified for any unit on the controller.

With the KDZ, a unit CSR can be specified only for the first unit set in the associated DZ multiplex group.

VECTOR *vector-address*
: Specifies the vector address of the line controller. You must use the vector address selected on the device or 0 if the address is unknown (range: 0 to 774 octal).

Examples:

```
CFE>DEFINE KNOWN LINES MAXIMUM DATA 256
```

This command establishes the maximum frame size for all known lines as 256 bytes.

```
CFE>DEFINE LINE DZ-3 SPEED 3600
```

This command sets the baud rate of line DZ-3 at 3600.

Use the DEFINE LOGGING command to create or modify logging parameters in the permanent data base.

Format:

```
DEFINE  ⎧KNOWN LOGGING  ⎫   ⎡⎧EVENTS list    ⎫⎤
        ⎨LOGGING CONSOLE⎬   ⎣⎩KNOWN EVENTS   ⎭⎦
        ⎨LOGGING FILE   ⎬   ⎡ STATE  ⎧OFF⎫⎤
        ⎩LOGGING MONITOR⎭   ⎣        ⎩ON ⎭⎦
```

where

KNOWN LOGGING	Indicates that the specified parameters are to be created or modified for all known logging components.
LOGGING CONSOLE	Indicates that the specified parameters are to be created or modified for the logging console only.
LOGGING FILE	Indicates that the specified parameters are to be created or modified for the logging file only.
LOGGING MONITOR	Indicates that the specified parameters are to be created or modified for the logging monitor program only.
EVENTS *list*	Specifies the event class and type(s) to be logged. See Chapter 2 for the event list format.
KNOWN EVENTS	Specifies that all events that DECnet-RSX can generate are to be logged.
STATE	Sets the operational state of the logging component when the system is loaded. When the state is OFF, events are discarded.

Example:

```
CFE>DEFINE LOGGING MONITOR STATE ON
```

This command changes the state of the logging monitor to ON.

CFE

DEFINE MODULE X25-ACCESS

P

Use the DEFINE MODULE X25-ACCESS command to specify a destination name for a remote DTE in the permanent data base. Destinations defined in the permanent data base apply to all terminals.

Format:

DEFINE MODULE X25-ACCESS DESTINATION *dest-name* NUMBER *dte-address*

where

DESTINATION *dest-name*	Specifies a destination name for the identified remote DTE address.
NUMBER *dte-address*	Specifies the remote DTE address to be associated with the specified destination name.

Example:

```
CFE>DEFINE MODULE X25-ACCESS DESTINATION PRS44 NUMBER 219746
```

This command establishes remote DTE 219746 as having the name PRS44.

Use the DEFINE MODULE X25-PROTOCOL command to create or modify DTE, group, or protocol module parameters in the permanent data base.

Format:

```
DEFINE MODULE X25-PROTOCOL  { { DTE dte-address     [CHANNELS list]              }
                            { { KNOWN DTES          [COUNTER TIMER seconds]      }
                            {                       [LINE line-id]               }
                            {                       [MAXIMUM CIRCUITS count]     }
                            {                       [STATE {OFF}]                }
                            {                       [      {ON }]                }
                            { { GROUP group-name    [DTE dte-address]     }
                            { { KNOWN GROUPS        [NUMBER group-number] }
                            {                       [TYPE BILATERAL]      }
                            {   [CALL TIMER seconds]
                            {   [CLEAR TIMER seconds]
                            {   [DEFAULT DATA byte-count]
                            {   [DEFAULT WINDOW block-count]
                            {   [MAXIMUM DATA byte-count]
                            {   [MAXIMUM CLEARS count]
                            {   [MAXIMUM RESETS count]
                            {   [MAXIMUM RESTARTS count]
                            {   [MAXIMUM WINDOW count]
                            {   [RESET TIMER seconds]
                            {   [RESTART TIMER seconds]
```

where

DTE-related parameters:

DTE *dte-address*	Defines parameters for the specified local DTE only. See *RSX-11 PSI Network-specific Information* for the address format on your network.
KNOWN DTES	Defines parameters for all local DTEs.
CHANNELS *list*	Specifies a list of logical channel numbers to be used for outgoing calls or to be taken by incoming calls. Use one or more 1- to 4-digit channel numbers in the range of 1 to 4095. Use a hyphen (-) to indicate ranges; use a comma (,) to separate numbers within the list. Channel numbers are used in the order in which they are listed.
COUNTER TIMER *seconds*	Sets a timer whose expiration causes a DTE counter logging event.

LINE *line-id*	Specifies a particular line for use by the protocol module.
MAXIMUM CIRCUITS *count*	Specifies the maximum number of circuits the module can use (range: 16 to 512).
STATE	Sets the local DTE's operational state to OFF or ON when the system is loaded.

Group-related parameters:

GROUP *group-name*	Defines parameters for the specified group only.
KNOWN GROUPS	Defines parameters for all known groups.
DTE *dte-address*	Identifies the local DTE associated with the specified group. This parameter must be used with a group number.
NUMBER *group-number*	Specifies a 2-digit closed user group (CUG) number or a 4-digit bilateral closed user group (BCUG) number (range: 0 to 9999; you can omit leading zeroes). This parameter must be used with a DTE address.
TYPE BILATERAL	Required when the specified group is bilateral (BCUG).

Protocol-related parameters:

CALL TIMER *seconds*	Sets a timer that starts when a request is transmitted to set up a virtual circuit and whose expiration clears the request if no response has been received (range: 0 to 255).
CLEAR TIMER *seconds*	Sets a timer that starts when a request is transmitted to clear a virtual circuit and whose expiration causes a retransmission of the clear request if no response has been received (range: 0 to 255).
DEFAULT DATA *byte-count*	Specifies the default packet size (in bytes) for all SVCs. This value must not exceed that specified for the MAXIMUM DATA parameter and must be a power of 2 (range: 16 to 1024).
DEFAULT WINDOW *block-count*	Specifies the default number of data packets that can be sent over an SVC before you have to wait for an acknowledgment (range: 1 to 7). This value must not exceed that specified for MAXIMUM WINDOW.
MAXIMUM DATA *byte-count*	Specifies the maximum packet size you expect to receive on any SVC. This value must be at least 5 bytes less than the maximum data size specified for PSI lines (see DEFINE LINE) and must be a power of 2 (range: 16 to 1024).

MAXIMUM CLEARS *count* — Specifies the maximum number of times that the protocol module can try clearing a virtual circuit (range: 0 to 255).

MAXIMUM RESETS *count* — Specifies the maximum number of times the protocol module can try resetting a virtual circuit (range: 0 to 255).

MAXIMUM RESTARTS *count* — Specifies the maximum number of times the protocol module can try restarting a virtual circuit (range: 0 to 255).

MAXIMUM WINDOW *count* — Specifies the maximum number of data packets that can be sent over an SVC before you have to wait for an acknowledgment (range: 1 to 7).

RESET TIMER *seconds* — Sets a timer that starts when a reset is transmitted and whose expiration causes a retransmission of the reset message if no response has been received (range: 0 to 255).

RESTART TIMER *seconds* — Sets a timer that starts when a restart is transmitted and whose expiration causes a retransmission of the restart message if no response has been received (range: 0 to 255).

Examples:

```
CFE>DEFINE MODULE X25-PROTOCOL KNOWN DTES STATE ON
```

This command sets the state of all local DTEs to ON when the system is loaded.

```
CFE>DEFINE MODULE X25-PROTOCOL MAXIMUM RESTARTS 20
```

This command limits to 20 the number of times the protocol module is to try restarting a virtual circuit.

DEFINE MODULE X25-SERVER DEFINE MODULE X29-SERVER

Use the DEFINE MODULE X25/X29-SERVER commands to create or modify server module parameters in the permanent data base.

Restriction:

If you specify CALL MASK or CALL VALUE, you must specify both, and the *hex-value* of each must contain the same number of digits.

Format:

```
DEFINE MODULE {X25-SERVER}  { [COUNTER TIMER seconds]
              {X29-SERVER}  { [MAXIMUM CIRCUITS count]
                            {
                            { {DESTINATION dest-name    [CALL MASK hex-value ] }
                            { {KNOWN DESTINATIONS       [CALL VALUE hex-value] }
                            { {                         [GROUP group-name]     }
                            { {                         [NUMBER dte-address]   }
                            { {                         [OBJECT object-id]     }
                            { {                         [PRIORITY priority]    }
                            { {                         [SUBADDRESSES range]   }
```

where

COUNTER TIMER *seconds*	Sets a timer whose expiration causes a server module counter logging event.
MAXIMUM CIRCUITS *count*	Specifies the maximum number of circuits that the module (that is, all destinations) can use (range: 1 to 255).
DESTINATION *dest-name*	Defines parameters for the specified destination only.
KNOWN DESTINATIONS	Defines parameters for all known destinations.
CALL MASK *hex-value*	Specifies the mask applied to incoming call data before it is tested against the call value (default: no mask). Use a hexadecimal string of 2 to 32 digits (must be an even number of digits); see restriction.
CALL VALUE *hex-value*	Specifies a string of information used to test incoming call data (default: no test string). Use a hexadecimal string of 2 to 32 digits (must be an even number of digits); see restriction.

GROUP *group-name*	Specifies the name of a closed user group (CUG) or a bilateral closed user group (BCUG) (default: no group).
NUMBER *dte-address*	Identifies the remote DTE that originates the call for the specified destination (default: no DTE).
OBJECT *object-id*	Specifies the task that runs when an incoming call activates it. Use either an object name or an object type code to identify the task (see Appendix B). (Required when creating a destination.)
PRIORITY *priority*	Specifies the priority of the destination. This is used to select one of a set of destinations for which the incoming call may be valid (range: 0 to 255, where 255 is the highest priority; default: 128).
SUBADDRESSES *range*	Specifies a range of local DTE subaddresses that identifies the destination for the incoming call.

Examples:

```
CFE>DEFINE MODULE X25-SERVER MAXIMUM CIRCUITS 20
```

This command limits the module to 20 circuits.

```
CFE>DEFINE MODULE X29-SERVER DESTINATION TKY029 PRIORITY 5
```

This command sets the priority of destination TKY029 to 5.

Use the DEFINE NODE command to create or modify parameters in the permanent data base for down-line loading and up-line dumping to the specified node (see Chapter 4). If you are not performing service operations on the node, NAME is the only meaningful parameter.

Restriction:

You must specify the NAME parameter if a node name has not previously been defined for the specified node.

Format:

```
DEFINE NODE node-id   [DIAGNOSTIC FILE file]
                      [DUMP ADDRESS address]
                      [DUMP COUNT number]
                      [DUMP FILE file]
                      [HARDWARE ADDRESS E-address]
                      [HOST node-id]
                      [LOAD FILE file]
                      [NAME node-name]
                      [SECONDARY [LOADER] file]
                      [SERVICE CIRCUIT circuit-id]
                      [SERVICE DEVICE device-type]
                      [SERVICE NODE VERSION {PHASEIII}]
                      [                     {PHASEIV }]
                      [[SERVICE] PASSWORD password]
                      [TERTIARY [LOADER] file]
```

where

NODE *node-id*	Specifies the node for which parameters are to be created or modified.
DIAGNOSTIC FILE *file*	(For Ethernet nodes only) Specifies the file to be read when the node is down-line loaded and requests diagnostics.
DUMP ADDRESS *address*	Specifies an octal address in memory at which to begin an up-line dump of the node.
DUMP COUNT *number*	Specifies the default number of memory units to up-line dump from the node.
DUMP FILE *file*	Specifies the file that is to receive a copy of the system at the time of the crash when the node is up-line dumped.

HARDWARE ADDRESS *E-address*	Identifies the Ethernet hardware address that was originally assigned to the DEUNA controller for the system on the node. This address is used during operations such as down-line load to communicate with the system before it has set up its physical address.
HOST *node-id*	Specifies a host node for all service operations (default: executor node).
LOAD FILE *file*	Specifies a file containing the system software for down-line loading to the node.
NAME *node-name*	Specifies a node name to be associated with the node ID.
SECONDARY LOADER *file*	Specifies a file containing secondary loader software for down-line loading to the node.
SERVICE CIRCUIT *circuit-id*	Specifies the circuit to be used for down-line loading and up-line dumping. This circuit is the default value for the VIA parameter of the LOAD command.
SERVICE DEVICE *device-type*	Specifies the node's line controller for the service line over which the operation is to take place. Possible devices are:

DA	DMC	DPV	DV	KDZ
DL	DMP	DU	DZ	QNA
DLV	DMV	DUP	KDP	UNA

SERVICE NODE VERSION	Specifies the node as a Phase III or Phase IV (default) node.
SERVICE PASSWORD *password*	Specifies the password required to trigger the bootstrap mechanism on the node.
TERTIARY LOADER *file*	Specifies a file containing tertiary loader software for down-line loading to the node.

Example:

```
CFE>DEFINE NODE 205 NAME DALLAS
```

This command identifies node 205 as DALLAS.

Use the DEFINE OBJECT command to create a new object or to modify object parameters in the permanent data base.

Format:

DEFINE OBJECT *type-code* [COPIES {*number* | SINGLE}]
[NAME *object-name*]
[USER {DEFAULT | LOGIN}]
[VERIFICATION {INSPECT | OFF | ON}]

where

OBJECT *type-code*	Specifies the object for which parameters are to be created or modified. Use an object type code from Appendix B.
COPIES *number*	Specifies the maximum number of copies of a task that can be run at once (range: 2 to 64). Use the keyword SINGLE (default) if the task is not multicopy.
NAME *object-name*	Specifies a name to be associated with the object. Use any valid RSX task name. If the task is multicopy, the name must end with $$$.
USER	Specifies the UIC under which a task is to run when the object's VERIFICATION option is set to ON or INSPECT.

DEFAULT	(Default) The task runs under the default UIC under which it was built or installed.
LOGIN	The task runs under the log-in UIC.

VERIFICATION	Specifies the degree of access to the controlled object.

INSPECT	Allows access to the object while indicating to it whether or not the access control information is valid.
OFF	(Default) Allows access to the object regardless of the validity of the access control information.
ON	Allows access to the object only for inbound connections with valid access control information.

Example:

```
CFE>DEFINE OBJECT 23 VERIFICATION ON
```

This command allows access to object 23 only to inbound connections with valid access control information.

Use the DEFINE PROCESS command to modify process parameters in the permanent data base.

Restriction:

Data space is allocated within the process's addressing space only.

Format:

```
DEFINE {PROCESS process-name } [MAXIMUM CONTROLLERS count]
       {KNOWN PROCESSES      } [MAXIMUM LINES number]
                               [STATE {CLEARED}]
                               [      {ON     }]
```

where

PROCESS *process-name*	Modifies parameters for the specified process only. Use a process name from the table in Chapter 2.
KNOWN PROCESSES	Modifies parameters for all known processes.
MAXIMUM CONTROLLERS *count*	Specifies the maximum number of hardware controllers (range: 1 to 64) for which the process is to allocate data space (see restriction).
MAXIMUM LINES *number*	Specifies the maximum number of lines (range: 1 to 64) for which the process is to allocate data space (see restriction).
STATE	Sets the operational state of the process when the system is loaded.

CLEARED	Not available; must be manually loaded.
ON	Automatically loaded and available.

Example:

```
CFE>DEFINE PROCESS DMC MAXIMUM CONTROLLERS 5
```

This command specifies that process DMC allocate data space for five controllers.

Use the DEFINE SYSTEM command to modify Communications Executive parameters in the permanent data base (see Chapter 5).

Format:

```
DEFINE SYSTEM   [LARGE BUFFER SIZE number]
                [[LOCATION] {FIRSTFIT }]
                [           {TOPDOWN  }]
                [MAXIMUM CONTROL BUFFERS number]
                [MAXIMUM LARGE BUFFERS number]
                [MAXIMUM SMALL BUFFERS number]
                [MINIMUM RECEIVE BUFFERS number]
                [POOL BYTE-AREA byte-count]
                [POOL NAME pool-name]
                [POOL PARTITION partition-name]
```

where

LARGE BUFFER SIZE *number*
: Specifies the size (in bytes) of large buffers (range: for DECnet, 192 to 4096; for PSI only, 128 to 4096).

LOCATION
: Specifies the type of dynamic memory allocation for any process whose state is ON in the permanent data base.

  FIRSTFIT — (Default) Loads the processes at the first available space that is large enough.

  TOPDOWN — Loads the processes at the top of the partition.

MAXIMUM CONTROL BUFFERS *number*
: Specifies the maximum number of control buffers available for Communications Executive system use (range: 1 to 200).

MAXIMUM LARGE BUFFERS *number*
: Specifies the maximum number of large buffers available for Communications Executive system use (range: 1 to 200).

MAXIMUM SMALL BUFFERS *number*
: Specifies the maximum number of small buffers available for Communications Executive system use (range: 1 to the maximum number of control buffers).

MINIMUM RECEIVE BUFFERS *number*
: Specifies the minimum number of receive buffers available for Communications Executive system use (range: 1 to *n*; *n* is the maximum number of large buffers minus 1).

POOL BYTE-AREA *byte-count*	Specifies the number of bytes needed in the network pool (range: 0 to 4095).
POOL NAME *pool-name*	Specifies the name of the subpartition to be created. Use a name consisting of 1 to 6 Radix-50 characters.
POOL PARTITION *partition-name*	Specifies the name of the partition into which the network pool is to be loaded. Use a name consisting of 1 to 6 Radix-50 characters.

Example:

```
CFE>DEFINE SYSTEM MAXIMUM SMALL BUFFERS 50
```

This command sets the maximum number of small buffers in the system to 50.

Use the KMX-DUMP command to dump KMX microcode to disk. For more information, see Chapter 9.

NOTE

> This command halts all lines on the KMX device. To bring the device back up after the dump, use the NCP SET LINE command to set the state of all lines first to OFF and then to ON.

Format:

KMX-DUMP LINE *line-id* [FILENAME *file*]

where

LINE *line-id*	Specifies a KMX line (see note above). Specify *line-id* in the form KMX-*c*-*u*.
FILENAME *file*	Names the file to hold the binary dump (default: LB:[1,6]PSIKMXMEM.SYS).

Example:

```
NCP>KMX-DUMP LINE KMX-0-1 LB:[1,6]KMXMC.DMP
```

This command dumps the microcode of KMX-0-1 to file LB:[1,6]KMXMC.DMP.

Use the LIST CIRCUIT command to display circuit information stored in the permanent data base (see Chapter 3).

Format:

LIST {CIRCUIT *circuit-id* | KNOWN CIRCUITS}

where

CIRCUIT *circuit-id*	Displays information for the specified circuit only.
KNOWN CIRCUITS	Displays information for all known circuits.

Example:

```
CFE>LIST CIRCUIT DMC-0
```

This command causes the system to display static information for circuit DMC-0.

Use the LIST EXECUTOR command to display executor node information stored in the permanent data base (see Chapter 3).

Format:

LIST EXECUTOR

Use the LIST LINE command to display line information stored in the permanent data base (see Chapter 3).

Format:

LIST $\left\{\begin{array}{l}\text{LINE } \textit{line-id} \\ \text{KNOWN LINES}\end{array}\right\}$

where

LINE *line-id*	Displays information for the specified line only.
KNOWN LINES	Displays information for all known lines.

Example:

```
CFE>LIST KNOWN LINES
```

This command causes the system to display static information for every line in the network.

Use the LIST LOGGING command to display logging information stored in the permanent data base (see Chapter 3).

Format:

```
LIST  { KNOWN LOGGING    }
      { LOGGING CONSOLE  }
      { LOGGING FILE     }
      { LOGGING MONITOR  }
```

where

KNOWN LOGGING	Displays information for all known logging components.
LOGGING CONSOLE	Displays information for the logging console only.
LOGGING FILE	Displays information for the logging file only.
LOGGING MONITOR	Displays information for the logging monitor program only.

Example:

```
CFE>LIST LOGGING MONITOR EVENTS
```

This command causes the system to display static information for the logging monitor program.

Use the LIST MODULE X25-ACCESS command to display remote DTE destination information stored in the permanent data base (see Chapter 3).

Format:

LIST MODULE X25-ACCESS { DESTINATION *dest-name* | KNOWN DESTINATIONS }

where

DESTINATION *dest-name*	Displays information for the specified destination only.
KNOWN DESTINATIONS	Displays information for all known destinations.

Example:

```
CFE>LIST MODULE X25-ACCESS DESTINATION NYC42
```

This command causes the system to display static information for destination NYC42.

LIST MODULE X25-PROTOCOL

Use the LIST MODULE X25-PROTOCOL command to display protocol module information stored in the permanent data base (see Chapter 3). The information may be qualified by either DTE or group.

Format:

$$\text{LIST MODULE X25-PROTOCOL} \quad \left[\begin{array}{l} \left\{ \begin{array}{l} \text{DTE } \textit{dte-address} \\ \text{KNOWN DTES} \end{array} \right\} \\ \left\{ \begin{array}{l} \text{GROUP } \textit{group-name} \\ \text{KNOWN GROUPS} \end{array} \right\} \end{array} \right]$$

where

DTE *dte-address*	Displays information for the specified DTE only.
KNOWN DTES	Displays information for all known DTEs.
GROUP *group-name*	Displays information for the specified group only.
KNOWN GROUPS	Displays information for all known groups.

Example:

```
CFE>LIST MODULE X25-PROTOCOL GROUP CUG98
```

This command causes the system to display static information for group CUG98.

Use the LIST MODULE X25/X29-SERVER commands to display server module information stored in the permanent data base (see Chapter 3). This information may be qualified by destination.

Format:

LIST MODULE {X25-SERVER | X29-SERVER} {DESTINATION *dest-name* | KNOWN DESTINATIONS}

where

DESTINATION *dest-name*	Displays information for the specified destination only.
KNOWN DESTINATIONS	Displays information for all known destinations.

Example:

```
CFE>LIST MODULE X29-SERVER DESTINATION PG49
```

This command causes the system to display static server module information for destination PG49.

Use the LIST NODE command to display node information stored in the permanent data base (see Chapter 3).

Format:

$$\text{LIST} \begin{Bmatrix} \text{NODE } \textit{node-id} \\ \text{KNOWN NODES} \end{Bmatrix}$$

where

NODE *node-id*	Displays information for the specified node only.
KNOWN NODES	Displays information for all known nodes.

Example:

```
CFE>LIST NODE BOS51
```

This command causes the system to display static information for node BOS51.

Use the LIST OBJECT command to display object information stored in the permanent data base (see Chapter 3).

Format:

LIST {OBJECT *type-code* | KNOWN OBJECTS}

where

OBJECT *type-code*	Displays information for the specified object only. Use an object type code from Appendix B.
KNOWN OBJECTS	Displays information for all known objects.

Example:

```
CFE>LIST KNOWN OBJECTS
```

This command causes the system to display static information for all objects known to the network.

Use the LIST PROCESS command to display process information stored in the permanent data base (see Chapter 3).

Format:

LIST {PROCESS *process-name* | KNOWN PROCESSES}

where

PROCESS *process-name*	Displays information for the specified process only. Use a process name from the table in Chapter 2.
KNOWN PROCESSES	Displays information for all known processes.

Example:

```
CFE>LIST PROCESS ADI
```

This command causes the system to display static process information.

Use the LIST SYSTEM command to display system information stored in the permanent data base (see Chapter 3).

Format:

LIST SYSTEM

Use the LOAD NODE command to down-line load a specified RSX-11S remote node (see Chapter 4).

Format:

LOAD NODE *node-id*
[ADDRESS *node-address*]
[FROM *file*]
[HOST *node-id*]
[NAME *node-name*]
[PHYSICAL ADDRESS *E-address*]
[SECONDARY [LOADER] *file*]
[SERVICE DEVICE *device-type*]
[SERVICE NODE VERSION {PHASEIII | PHASEIV}]
[[SERVICE] PASSWORD *password*]
[TERTIARY [LOADER] *file*]
[VIA *circuit-id*]

where

NODE *node-id*	Specifies the node to be down-line loaded.
ADDRESS *node-address*	Specifies the address that the node is to use when it comes up.
FROM *file*	Specifies the file specification of the load file containing the system software to be down-line loaded.
HOST *node-id*	Specifies the default host that the node is to use when it comes up.
NAME *node-name*	Specifies the node name that the node is to use when it comes up.
PHYSICAL ADDRESS *E-address*	(For Ethernet nodes only) Identifies the Ethernet physical address that the node currently uses to identify itself. (Required for Ethernet circuits if the hardware address parameter has not been specified in the volatile data base; see SET NODE.)

SECONDARY LOADER *file*	Specifies the file specification of a secondary load program to be down-line loaded.
SERVICE DEVICE *device-type*	Specifies the node's line controller for the service line over which the operation is to take place. Possible devices are:

DA	DMC	DPV	DV	KDZ
DL	DMP	DU	DZ	QNA
DLV	DMV	DUP	KDP	UNA

SERVICE NODE VERSION	Specifies the node as a Phase III or Phase IV (default) node.
SERVICE PASSWORD *password*	Specifies the password required to trigger the bootstrap mechanism on the node.

NOTE

If you do not want a password echoed to your terminal while you enter it, press the RETURN key after the keyword PASSWORD. NCP prompts for the password and turns off echoing until the next prompt.

TERTIARY LOADER *file*	Specifies the file specification of a third load program to be down-line loaded.
VIA *circuit-id*	Specifies the circuit over which the load is to take place.

Examples:

```
NCP>LOAD NODE ATL25
```

This command loads node ATL25. Default load parameters are obtained from the volatile data base.

```
NCP>LOAD NODE ATL25 VIA DMC-1
```

This command loads node ATL25 by way of circuit DMC-1.

Use the LOAD VIA command to down-line load an RSX-11S remote node via the specified circuit (see Chapter 4).

Format:

LOAD VIA *circuit-id* [ADDRESS *node-address*]
[FROM *file*]
[HOST *node-id*]
[NAME *node-name*]
[PHYSICAL ADDRESS *E-address*]
[SECONDARY [LOADER] *file*]
[SERVICE DEVICE *device-type*]
[SERVICE NODE VERSION {PHASEIII | PHASEIV}]
[[SERVICE] PASSWORD *password*]
[TERTIARY [LOADER] *file*]

where

VIA *circuit-id*	Specifies a circuit over which the load is to take place.
ADDRESS *node-address*	Specifies the address the node is to use when it comes up.
FROM *file*	Specifies the file specification of the load file containing the system software to be down-line loaded.
HOST *node-id*	Specifies the default host that the node is to use when it comes up.
NAME *node-name*	Specifies the node name that the node is to use when it comes up.
PHYSICAL ADDRESS *E-address*	(For Ethernet nodes only) Identifies the Ethernet physical address that the node currently uses to identify itself. (Required for Ethernet circuits if the hardware address parameter has not been specified in the volatile data base; see SET NODE.)
SECONDARY LOADER *file*	Specifies the file specification of a secondary load program to be down-line loaded.
SERVICE DEVICE *device-type*	Specifies the node's line controller for the service line over which the operation is to take place. Possible devices are:

DA	DMC	DPV	DV	KDZ
DL	DMP	DU	DZ	QNA
DLV	DMV	DUP	KDP	UNA

SERVICE NODE VERSION	Specifies the node as a Phase III or Phase IV (default) node.
SERVICE PASSWORD *password*	Specifies the password required to trigger the bootstrap mechanism on the node.

NOTE

If you do not want a password echoed to your terminal while you enter it, press the RETURN key after the keyword PASSWORD. NCP prompts for the password and turns off echoing until the next prompt.

TERTIARY LOADER *file*	Specifies the file specification of a third load program to be down-line loaded.

Example:

```
NCP>LOAD VIA DMC-0
```

This command loads the node connected to the executor node via circuit DMC-0.

Use the LOOP CIRCUIT command to test a DECnet or Ethernet circuit in the network. Use the LOOP LINE command to test a PSI line in the network. Each command causes test blocks of data to be transmitted over the specified line or circuit. See Chapter 3 for more information on loopback testing.

When testing an Ethernet circuit, you can use the HELP parameter to specify a third node to assist with the test. If you specify the HELP parameter, you must also specify the PHYSICAL ADDRESS and ASSISTANT PHYSICAL ADDRESS parameters. (**Note:** They cannot be multicast addresses.) If you do not know the addresses, you can use the NODE and ASSISTANT NODE parameters instead.

Format:

```
LOOP { LINE line-id
     { CIRCUIT circuit-id [ HELP { FULL     }                                  ] }
     {                    [      { RECEIVE  }                                  ] }
     {                    [      { TRANSMIT }                                  ] }
     {                    [ { [PHYSICAL ADDRESS E-address                   ] } ] }
     {                    [ { [ASSISTANT PHYSICAL ADDRESS E-address         ] } ] }
     {                    [ { [NODE E-node-name             ]               } ] }
     {                    [ { [ASSISTANT NODE E-node-name   ]               } ] }
     [COUNT count]
     [LENGTH length]
     [WITH { MIXED  } ]
     [     { ONES   } ]
     [     { ZEROES } ]
```

where

LINE *line-id*	Specifies a PSI line for loopback testing.
CIRCUIT *circuit-id*	Specifies a circuit for loopback testing.

HELP *help-type*	Specifies the degree to which a third node is to assist with an Ethernet loop circuit test. FULL — The assisting node is to both receive and transmit the test packet (see the example). RECEIVE — The assisting node is only to receive the test packet. TRANSMIT — The assisting node is only to transmit the test packet. If you specify HELP, you must also specify either the PHYSICAL ADDRESS and ASSISTANT PHYSICAL ADDRESS parameters or, if the addresses are not known, the NODE and ASSISTANT NODE parameters.
PHYSICAL ADDRESS *E-address*	Specifies the physical address (not a multicast address) of an Ethernet node that is to be the destination of a loop test message.
ASSISTANT PHYSICAL ADDRESS *E-address*	Specifies the physical address (not a multicast address) of an Ethernet node that is to assist in the loop circuit test.
NODE *E-node-name*	Specifies the name of an Ethernet node that is to be the destination of a loop test message.
ASSISTANT NODE *E-node-name*	Specifies the name of an Ethernet node that is to assist in the loop circuit test.
COUNT *count*	Specifies the number of blocks to be sent during loopback testing (range: 1 to 65535; default: 1 block is looped).
LENGTH *length*	Specifies the length (in bytes) of blocks to be sent during loopback testing. When testing over the Ethernet, the length must be a decimal integer in the range of 32 to 1484. Otherwise, the length must be a decimal integer in the range of 1 to *n*, where *n* must be less than the smaller of either the local looper buffer size or the remote mirror buffer size (default: 40 bytes).
WITH	Specifies the type of binary information to be sent during testing (default: MIXED — a combination of ones and zeroes).

Example:

```
NCP>LOOP CIRCUIT UNA-0 HELP FULL PHYSICAL ADDRESS AA-00-04-00-F9-04 -<RET>
NCP>ASSISTANT PHYSICAL ADDRESS AA-00-04-00-04-A9
```

This command (shown in continuation format) tests circuit UNA-0 with the assistance of the node specified in ASSISTANT PHYSICAL ADDRESS. The initiating node sends a test packet to the assisting node. The assisting node processes the packet and passes the packet to the destination node specified in PHYSICAL ADDRESS. The destination node receives the packet and transmits the packet back to the assisting node. The assisting node then returns the packet to the initiating node.

Use the LOOP NODE command to test a node in the network. You can include access control information if the node requires it. If you are testing the executor node, you can use the LOOP EXECUTOR command. Either command causes test blocks of data to be transmitted to the specified node. See Chapter 3 for more information on loopback testing.

Format:

```
LOOP {NODE node-id[acc-con-info]}  [COUNT count]
     {EXECUTOR                  }  [LENGTH length]
                                   [WITH {MIXED }]
                                   [     {ONES  }]
                                   [     {ZEROES}]
```

where

NODE *node-id*	Specifies a node for loopback testing.
acc-con-info	Specifies access control information, if required.
EXECUTOR	Specifies the executor node for loopback testing.
COUNT *count*	Specifies the number of blocks to be sent during loopback testing (range: 1 to 65535; default: 1 block is looped).
LENGTH *length*	Specifies the length (in bytes) of blocks to be sent during loopback testing. When testing over the Ethernet, the length must be a decimal integer in the range of 32 to 1484. Otherwise, the length must be a decimal integer in the range of 1 to *n*, where *n* must be less than the smaller of either the local looper buffer size or the remote mirror buffer size (default: 40 bytes).
WITH	Specifies the type of binary information to be sent during testing (default: MIXED — a combination of ones and zeroes).

Example:

```
NCP>LOOP NODE NYC COUNT 10
```

This command loops 10 blocks of mixed test messages to remote node NYC. Each block is 40 bytes.

Use the PURGE CIRCUIT command to zero the specified PSI circuit counter timer(s) in the permanent data base.

Format:

PURGE $\left\{\begin{array}{l}\textbf{CIRCUIT } \textbf{\textit{circuit-id}} \\ \textbf{KNOWN CIRCUITS}\end{array}\right\}$ **COUNTER TIMER**

where

CIRCUIT *circuit-id*	Zeroes the counter timer for the specified circuit only.
KNOWN CIRCUITS	Zeroes the counter timers for all known circuits.

Example:

```
CFE>PURGE KNOWN CIRCUITS COUNTER TIMER
```

This command zeroes the counter timers for all known PSI circuits.

Use the PURGE LINE command to zero the specified PSI line counter timer(s) in the permanent data base.

Format:

PURGE $\left\{\begin{array}{l}\textbf{LINE } \textit{line-id} \\ \textbf{KNOWN LINES}\end{array}\right\}$ **COUNTER TIMER**

where

LINE *line-id*	Zeroes the counter timer for the specified line only.
KNOWN LINES	Zeroes the counter timers for all known lines.

Example:

```
CFE>PURGE LINE SDP-0 COUNTER TIMER
```

This command zeroes the counter timer for line SDP-0.

Use the PURGE LOGGING command to remove the specified logging event parameters from the permanent data base.

Format:

```
PURGE {KNOWN LOGGING   } {ALL EVENTS   }
      {LOGGING CONSOLE } {EVENTS list  }
      {LOGGING FILE    } {KNOWN EVENTS }
      {LOGGING MONITOR }
```

where

KNOWN LOGGING	Removes parameters for all known logging components.
LOGGING CONSOLE	Removes parameters for logging to the console.
LOGGING FILE	Removes parameters for logging to a file.
LOGGING MONITOR	Removes parameters for logging to a monitor program.
ALL EVENTS	Removes parameters for all logging event classes and types.
EVENTS *list*	Removes parameters for the logging event class and type(s) specified in *list*. See Chapter 2 for the event list format.
KNOWN EVENTS	Removes parameters for all logging events that DEC-net-RSX can generate.

Example:

```
CFE PURGE KNOWN LOGGING ALL EVENTS
```

This command removes all classes and types of events for all known logging components.

Use the PURGE MODULE X25-ACCESS command to remove parameters for the specified destination(s) from the permanent data base.

Format:

PURGE MODULE X25-ACCESS $\left\{\begin{array}{l}\textbf{DESTINATION}\ \textbf{\textit{dest-name}}\\ \textbf{KNOWN DESTINATIONS}\end{array}\right\}$ **ALL**

where

DESTINATION *dest-name*	Removes parameters for the specified destination only.
KNOWN DESTINATIONS	Removes parameters for all known destinations.
ALL	(Required) Removes all parameters for the specified destination(s).

Example:

```
CFE>PURGE MODULE X25-ACCESS DESTINATION DALLAS ALL
```

This command removes destination DALLAS and all associated parameters.

PURGE MODULE X25-PROTOCOL

Use the PURGE MODULE X25-PROTOCOL command to zero the counter timer(s) for the specified DTE(s) or to remove all parameters for the specified DTE(s) or group(s) from the permanent data base.

Format:

```
PURGE MODULE X25-PROTOCOL  { {DTE dte-address  }  {ALL          } }
                           { {KNOWN DTES       }  {COUNTER TIMER} }
                           { {GROUP group-name }   ALL            }
                           { {KNOWN GROUPS     }                  }
```

where

DTE *dte-address*	Clears the specified DTE only. Be aware that purging a DTE removes all PVCs and CUGs associated with it and purges them from the permanent data base.
KNOWN DTES	Clears all known DTEs.
GROUP *group-name*	Removes parameters for the specified group only.
KNOWN GROUPS	Removes parameters for all known groups.
ALL	Removes all parameters for the specified DTE(s) or group(s).
COUNTER TIMER	Zeroes the counter timer(s) for the specified DTE(s).

Example:

```
CFE>PURGE MODULE X25-PROTOCOL GROUP CUG111 ALL
```

This command removes parameters for group CUG111.

PURGE MODULE X25–SERVER PURGE MODULE X29–SERVER

Use the PURGE MODULE X25/X29-SERVER commands to zero the counter timer or to remove parameters for the specified destination(s) from the permanent data base.

Format:

```
PURGE MODULE {X25-SERVER}  [ COUNTER TIMER                                   ]
             {X29-SERVER}  [                                                 ]
                           [ {DESTINATION dest-name}  ALL                    ]
                           [ {KNOWN DESTINATIONS   }                         ]
```

where

COUNTER TIMER	Zeroes the server module counter timer.
DESTINATION *dest-name*	Removes parameters for the specified destination only.
KNOWN DESTINATIONS	Removes parameters for all known destinations.
ALL	(Required when you specify a destination parameter) Removes all parameters for the specified destination(s).

Example:

```
CFE PURGE MODULE X25-SERVER DESTINATION ALBANY ALL
```

This command removes destination ALBANY and all associated parameters.

Use the PURGE NODE command to remove specified node parameters from the permanent data base on the local node.

Format:

```
PURGE {NODE node-id   } [ALL]
      {KNOWN NODES    } [DIAGNOSTIC FILE]
                        [DUMP ADDRESS]
                        [DUMP COUNT]
                        [DUMP FILE]
                        [HARDWARE ADDRESS]
                        [HOST]
                        [LOAD FILE]
                        [SECONDARY [LOADER]]
                        [SERVICE CIRCUIT]
                        [SERVICE DEVICE]
                        [[SERVICE] PASSWORD]
                        [TERTIARY [LOADER]]
```

where

NODE *node-id*	Removes parameters for the specified node only.
KNOWN NODES	Removes parameters for all known nodes.
ALL	Removes all parameters for the specified node(s). If you specify ALL, you cannot include any other parameters.
DIAGNOSTIC FILE	Removes the identification of the down-line load diagnostics file.
DUMP ADDRESS	Removes the node's up-line dump address.
DUMP COUNT	Removes the up-line dump count.
DUMP FILE	Removes the up-line dump file identification.
HARDWARE ADDRESS	Removes the Ethernet address of the system hardware.
HOST	Removes the host node identification.
LOAD FILE	Removes the down-line load file identification.

SECONDARY LOADER	Removes the secondary down-line load file identification.
SERVICE CIRCUIT	Removes the circuit parameter associated with the node for down-line load operations.
SERVICE DEVICE	Removes the service device type.
SERVICE PASSWORD	Removes the password parameter required to trigger the bootstrap mechanism.
TERTIARY LOADER	Removes the tertiary down-line load file identification.

Example:

```
CFE>PURGE NODE BOS ALL
```

This command removes loop node BOS from the data base.

Use the PURGE OBJECT command to remove all parameters for the specified object(s) from the permanent data base.

Format:

PURGE { OBJECT *object-type* | KNOWN OBJECTS }

where

OBJECT *object-type*	Removes parameters for the specified object only. Use an object type code from Appendix B.
KNOWN OBJECTS	Removes parameters for all known objects.

Example:

```
CFE>PURGE OBJECT 200
```

This command removes all parameters for object 200.

Use the NCP SET ALIAS command to specify an alias name for a node in the volatile data base. Use the VNP command to do the same with the system image file.

Format:

SET ALIAS *alias-name* DESTINATION *dest-node* [[SCOPE] *scope*]

where

ALIAS *alias-name*	Specifies the alias name to be assigned to the destination node. *Alias-name* consists of 1 to 6 alphanumeric characters, including at least 1 alphabetic character.
DESTINATION *dest-node*	Specifies the destination node and any access control parameters associated with this alias for that node. See Chapter 2 for the access control format.
SCOPE *scope*	See definition in Section 6.5.2.

Example:

```
NCP>SET ALIAS DORCH5 DESTINATION BOS27/DAVIS/SNAIL - RET
NCP>SCOPE TERMINAL TT22:
```

This command (shown in continuation format) sets alias DORCH5 to terminal TT22: on node BOS27. DAVIS is the account name, and SNAIL is the password.

Use the NCP SET CIRCUIT command to create or modify specified circuit parameters in the volatile data base. Use the VNP command to do the same with the system image file.

Restriction:

The VNP SET CIRCUIT command is invalid for PSI circuits.

Format:

```
SET {CIRCUIT circuit-id }   [COST cost]
    {KNOWN CIRCUITS     }   [COUNTER TIMER seconds]
                            [HELLO TIMER seconds]
                            [MULTIPOINT ACTIVE active-ratio]
                            [OWNER {DLX}]
                            [      {XPT}]
                            [SERVICE {DISABLE}]
                            [        {ENABLE }]
                            [STATE {OFF    }]
                            [      {ON     }]
                            [      {SERVICE}]
                            [TRIBUTARY  trib-address]
```

where

CIRCUIT *circuit-id*	Sets parameters for the specified circuit only.
KNOWN CIRCUITS	Sets parameters for all known circuits.
COST *cost*	Specifies the routing cost of the circuit (range: 1 to 25). Messages will travel between nodes along the path with the smallest total cost.
COUNTER TIMER *seconds*	(Valid for X25-SERVER circuits only; invalid for VNP) Sets a timer whose expiration causes a PSI circuit counter logging event.
HELLO TIMER *seconds*	Specifies the frequency of routing hello messages sent to adjacent nodes over the circuit.
MULTIPOINT ACTIVE *active-ratio*	(Valid only for circuits with multipoint software) Specifies the rate at which a tributary in the active multipoint state is polled (range: 1 to 255).
OWNER	Specifies the circuit owner.

DLX	DLX. (See the *DECnet-RSX Programmer's Reference Manual.*)
XPT	DECnet

SERVICE	Specifies whether the circuit is enabled or disabled for down-line loading and loopback testing.
STATE	Sets the circuit's operational state. For circuits owned by the executor, the state of the corresponding lines must be ON.

OFF	Not available.
ON	Available for normal use.
SERVICE	Available for service operations only.

TRIBUTARY *trib-address*	Specifies the data link physical tributary address of the multipoint circuit (range: 1 to 255).

Example:

```
NCP>SET CIRCUIT KDZ-1-3 STATE OFF
```

This command sets the state of circuit KDZ-1-3 to OFF.

Use the NCP SET EXECUTOR command to create or modify executor node parameters in the volatile data base. Use the VNP command to do the same with the system image file.

Format:

```
SET EXECUTOR   [HOST node-id]
               [RECEIVE PASSWORD password]
               [ROUTING TIMER seconds]
               [SEGMENT BUFFER SIZE number]
               [STATE {OFF                        }]
               [      {ON  [{FIXED  }]            }]
               [      {    [{UNFIXED}]            }]
               [      {SHUT                       }]
               [SUBADDRESSES range]
               [TRANSMIT PASSWORD password]
               [VERIFICATION [STATE] {OFF}]
               [                     {ON }]
```

where

HOST *node-id*	Specifies the host node identification for down-line load, trigger, and up-line dump.
RECEIVE PASSWORD *password*	Specifies a 1- to 8-character ASCII password that the executor expects to receive during initialization with an adjacent node.
ROUTING TIMER *seconds*	Sets a timer whose expiration forces a routing update on non-Ethernet circuits.
SEGMENT BUFFER SIZE *number*	Specifies the maximum size of transmit buffers (in bytes), thereby controlling the size of the NSP message segment to be sent. This value is the maximum size message that the End Communications layer can transmit; it does not include routing or data link overhead (range: 1 to 4096; default: 18 bytes less than the large buffer size specified in the DEFINE SYSTEM command). The segment buffer size can never be greater than the large buffer size minus 18 bytes.

STATE	Sets the operational state of the executor node. This parameter cannot be specified if you use the TELL prefix. OFF — Allows no new logical links; terminates existing links; forces the release of all mailboxes as quickly as possible. ON — Allows normal logical link activity. When using VNP to set the executor state to ON for DECnet-11S systems, you can specify one of the following task states for NTINIT after it has initialized the node: FIXED — (Default) NTINIT remains in memory. UNFIXED — NTINIT is removed from memory. SHUT — (Invalid for VNP) Allows no new logical links; does not destroy existing links; goes to the OFF state when all logical links terminate and all mailboxes are released.
SUBADDRESSES *range*	(Invalid for VNP) Specifies a range of local DTE subaddresses that are valid on any PSI circuit for incoming calls to the executor node.
TRANSMIT PASSWORD *password*	Specifies a 1- to 8-character ASCII password that the executor sends to the adjacent node during a node initialization sequence.
VERIFICATION STATE	Sets the logical link access control verification state for the executor node. OFF — The executor node does not verify access control on any incoming connect requests. ON — The executor node verifies access control on all incoming connect requests according to the options set for each object (see the discussion of access control in Chapter 2).

Example:

```
NCP>SET EXECUTOR HOST 12
```

This command sets node 12 as the host node for down-line load, trigger, and up-line dump operations.

Use the SET EXECUTOR NODE command to set a default executor for all NCP commands. This causes subsequent remotely executable NCP commands to be executed at the specified destination. Appendix A lists all NCP commands and flags those that are not remotely executable.

Restriction:

You cannot use the TELL prefix with this command (that is, this command is not remotely executable).

Format:

SET EXECUTOR NODE *node-id*[*acc-con-info*]

where

NODE *node-id*	Specifies the node (by address, alias, or name) that is to serve as the executor for subsequent NCP commands.
acc-con-info	Specifies access control information (if required by the node).

Example:

```
NCP>SET EXECUTOR NODE ATL25/NELSON/PAT
```

This command sets node ATL25 (user NELSON, password PAT) to executor status. Future commands will be sent to ATL25 for execution.

Use the NCP SET KNOWN LINES command either to load all known lines and create their associated parameters in the volatile data base or to create or modify parameters in the data base for all known lines that are already loaded. Use the VNP command to do the same with the system image file.

Restriction:

Except for STATE, the options allowed when you are loading lines are different from those used to create or modify parameters for lines that are already loaded. Any lines that you are loading must be in the CLEARED state. Lines that are already loaded must be in the ON or OFF states (that is, not CLEARED).

Format:

SET KNOWN LINES $\left[\textbf{STATE} \begin{Bmatrix}\textbf{OFF}\\ \textbf{ON}\end{Bmatrix}\right]$

Loading options:

[ALL]

[DEAD TIMER *milliseconds*]

[DELAY TIMER *milliseconds*]

$\left[\text{DUPLEX} \begin{Bmatrix}\text{FULL}\\ \text{HALF}\end{Bmatrix}\right]$

$\left[\text{[LOCATION]} \begin{Bmatrix}\text{FIRSTFIT}\\ \text{TOPDOWN}\end{Bmatrix}\right]$

Loaded options:

[COUNTER TIMER *seconds*]

$\left[\text{OWNER} \begin{Bmatrix}\text{DLX}\\ \textbf{PLI}\end{Bmatrix}\right]$

where

STATE
: (Invalid for VNP) Sets the operational state for all PSI lines to either OFF or ON.

Loading options:

ALL
: Loads the lines with all of the default parameters specified in the permanent data base. If you specify ALL, you cannot include any other parameters.

DEAD TIMER *milliseconds*
: (For DDCMP CONTROL lines only) Specifies the polling interval for a dead tributary.

DELAY TIMER *milliseconds*
: (For DDCMP CONTROL lines only) Specifies the minimum time to delay between polls in order to limit the effect of a fast control station on a slow tributary.

DUPLEX	(Not applicable for PSI or Ethernet lines) Sets the hardware transmission mode to full duplex or half duplex.
LOCATION	Specifies the type of dynamic memory allocation. FIRSTFIT — (Default) Loads the driver processes at the first available space that is large enough in the partition. TOPDOWN — (Invalid for VNP) Loads the driver processes at the top of the partition.

Loaded options:

COUNTER TIMER *seconds*	(Invalid for VNP) Sets a timer whose expiration causes a PSI line counter logging event.
OWNER	Specifies line ownership. DLX — DLX (See the *DECnet-RSX Programmer's Reference Manual.*) PLI — PSI (Invalid for VNP)

Example:

```
NCP>SET KNOWN LINES DUPLEX FULL
```

This command sets all lines known to the system to full duplex.

P

Use the NCP SET LINE command either to load the specified line and create its associated parameters in the volatile data base or to create or modify parameters in the data base for a line that is already loaded. Use the VNP command to do the same with the system image file.

Restriction:

Except for STATE, the options allowed when you are loading a line are different from those used to create or modify parameters for a line that is already loaded. Any line that you are loading must be in the CLEARED state. A line that is already loaded must be in the ON or OFF state (that is, not CLEARED).

Format:

SET LINE *line-id* [STATE {OFF | ON}]

Loading options:

[ALL]
[[CONTROLLER] CSR *csr-address*]
[DEAD TIMER *milliseconds*]
[DELAY TIMER *milliseconds*]
[DUPLEX {FULL | HALF}]
[[LOCATION] {FIRSTFIT | TOPDOWN}]
[MULTIPOINT DEAD *dead-ratio*]
[PRIORITY *hardware-priority*]
[UNIT CSR *csr-address*]
[VECTOR *vector-address*]

Loaded options:

[CONTROLLER {LOOPBACK | NORMAL}]
[COUNTER TIMER *seconds*]
[OWNER {DLX | PLI}]

where

LINE *line-id* — Specifies the line to be loaded and for which parameters are to be created or modified.

STATE — (Invalid for VNP) Sets a PSI line's operational state to OFF or ON.

Loading options:

ALL	Loads the line with all of the default parameters specified in the permanent data base. If you specify ALL, you cannot include any other parameters.
CONTROLLER CSR *csr-address*	Specifies the address of the first CSR for the line controller.
DEAD TIMER *milliseconds*	(For DDCMP CONTROL lines only) Specifies the polling interval for a dead tributary.
DELAY TIMER *milliseconds*	(For DDCMP CONTROL lines only) Specifies the minimum time to delay between polls in order to limit the effect of a fast control station on a slow tributary.
DUPLEX	(Not applicable for PSI or Ethernet lines) Sets the hardware transmission mode to full duplex or half duplex.
LOCATION	Specifies the type of dynamic memory allocation. FIRSTFIT — (Default) Loads the driver process at the first available space that is large enough in the partition. TOPDOWN — (Invalid for VNP) Loads the driver process at the top of the partition.
MULTIPOINT DEAD *dead-ratio*	Specifies the polling rate for a tributary in the dead multipoint state (range: 1 to 255).
PRIORITY *hardware-priority*	Specifies the hardware priority of the controller (range: 4 to 7).
UNIT CSR *csr-address*	Specifies the address of the first CSR for a DECnet device controlled by a KMC-11.

NOTE

With the KDP, a unit CSR can be specified for any unit on the controller.

With the KDZ, a unit CSR can be specified only for the first unit set in the associated DZ multiplex group.

VECTOR *vector-address*	Specifies the vector address of the line controller. You must use the vector address selected on the device or 0 if the address is unknown (range: 0 to 774 octal).

Loaded options:

CONTROLLER — (For Ethernet, DMC/DMR, DMP, and DMV lines; invalid for VNP) Specifies the controller mode for the line. The line's circuit must be in the OFF state.

LOOPBACK	Internal device loopback mode
NORMAL	Normal operating mode

COUNTER TIMER *seconds* — (Invalid for VNP) Sets a timer whose expiration causes a PSI line counter logging event.

OWNER — Specifies line ownership.

DLX	DLX (See the *DECnet-RSX Programmer's Reference Manual.*)
PLI	PSI (Invalid for VNP)

Example:

```
NCP SET LINE DMC-0 OWNER XPT
```

This command sets the owner of line DMC-0 as XPT, which assigns the line for DECnet operations only.

Use the NCP SET LOGGING command to create or modify logging parameters in the volatile data base. Use the VNP command to do the same with the system image file.

Restriction:

Whenever a circuit, line, module, node, or sink is specified in a SET LOGGING command, an event list or KNOWN EVENTS parameter must also be included.

Format:

```
SET {KNOWN LOGGING    } [ NAME name                                                      ]
    {LOGGING CONSOLE  } [ STATE {OFF}                                                    ]
    {LOGGING FILE     } [       {ON }                                                    ]
    {LOGGING MONITOR  } [                                                                ]
                        [ {EVENTS list    [[{CIRCUIT circuit-id                       }]]} ]
                        [ {KNOWN EVENTS   [[{LINE line-id                             }]]} ]
                        [ {               [[{MODULE {X25-PROTOCOL}                    }]]} ]
                        [ {               [[{       {X25-SERVER  }                    }]]} ]
                        [ {               [[{       {X29-SERVER  }                    }]]} ]
                        [ {               [[{NODE node-id                             }]]} ]
                        [ {               [[ SINK {EXECUTOR              }             ]]} ]
                        [ {               [[      {NODE {node-id}        }             ]]} ]
                        [ {               [[      {     {$HOST  }        }             ]]} ]
```

where

KNOWN LOGGING	Indicates that the specified parameters are to be created or modified for all known logging components.
LOGGING CONSOLE	Indicates that the specified parameters are to be created or modified for the logging console only.
LOGGING FILE	Indicates that the specified parameters are to be created or modified for the logging file only.
LOGGING MONITOR	Indicates that the specified parameters are to be created or modified for the logging monitor program only.
NAME *name*	Specifies the name of the console (default: CO0:), file (default: LB:[1,6]EVENTLOG.SYS), or monitor program (default: MON...) to which events are to be logged.
STATE	Sets the operational state of the logging component on the executor node. When the state is OFF, events are discarded.
EVENTS *list*	Specifies the event class and type(s) to be logged. See Chapter 2 for the event list format.

KNOWN EVENTS	Specifies that all events that DECnet-RSX can generate are to be logged.
CIRCUIT *circuit-id*	(Invalid for VNP) Logs the specified event(s) occurring on the specified circuit (see restriction).
LINE *line-id*	Logs the specified event(s) occurring on the specified line (see restriction).
MODULE	(Invalid for VNP) Logs the specified event(s) occurring on the specified module (see restriction).
NODE *node-id*	Logs the specified event(s) occurring on the specified node (see restriction).
SINK	Identifies the node where the specified event(s) are to be logged (see restriction).

EXECUTOR	(Default) Executor node
NODE *node-id*	The specified remote node
NODE $HOST	(Invalid for NCP) Host node

Examples:

```
NCP>SET KNOWN LOGGING KNOWN EVENTS CIRCUIT DMC-0 SINK NODE BOS-
TON
```

This command dictates that known events on circuit DMC-0 be sent to node BOSTON.

```
NCP>SET LOGGING CONSOLE EVENT 2.1 SINK NODE ATL
```

This command dictates that any occurrence of event 2.1 be logged on the console at node ATL.

Use the SET MODULE X25-ACCESS command to specify a destination name for a remote DTE in the volatile data base.

Format:

SET MODULE X25-ACCESS DESTINATION ***dest-name*** **NUMBER** ***dte-address*** **[[SCOPE]** ***scope*****]**

where

DESTINATION *dest-name*	Specifies a destination name for the identified remote DTE address.
NUMBER *dte-address*	Specifies the remote DTE address to be associated with the specified destination name.
SCOPE *scope*	See definition in Section 6.5.2.

Example:

```
NCP>SET MODULE X25-ACCESS DESTINATION RDG724 NUMBER 41106700200 - RET
NCP>SCOPE TERMINAL TT51:
```

This command (shown in continuation format) sets a destination name of RDG724 for terminal TT51: at DTE address 41106700200.

Use the SET MODULE X25-PROTOCOL command to create or modify DTE or group parameters in the volatile data base.

Format:

```
SET MODULE X25-PROTOCOL { {DTE dte-address}   [COUNTER TIMER seconds]   }
                        { {KNOWN DTES     }   [STATE {OFF }]            }
                        {                     [      {ON  }]            }
                        {                     [      {SHUT}]            }
                        { {GROUP group-name}  [DTE dte-address      ]   }
                        { {KNOWN GROUPS    }  [NUMBER group-number  ]   }
                        {                     [TYPE BILATERAL]          }
```

where

DTE-related parameters:

DTE *dte-address* — Sets parameters for the specified local DTE only. See *RSX-11 PSI Network-specific Information* for the address format on your network.

KNOWN DTES — Sets parameters for all local DTEs.

COUNTER TIMER *seconds* — Sets a timer whose expiration causes a DTE counter logging event.

STATE — Sets the local DTE's operational state.

- OFF — Not available.
- ON — Available for normal use.
- SHUT — Not available for new virtual circuit connections; turns off when the last virtual circuit disconnects.

Group-related parameters:

GROUP *group-name* — Sets parameters for the specified group only.

KNOWN GROUPS — Sets parameters for all known groups.

DTE *dte-address* — Specifies the local DTE associated with the specified group. This parameter must be used with a group number.

NUMBER *group-number*	Specifies a 2-digit closed user group (CUG) number or a 4-digit bilateral closed user group (BCUG) number (range: 0 to 9999; you can omit leading zeroes). This parameter must be used with a DTE address.
TYPE BILATERAL	Required when the specified group is bilateral (BCUG).

Example:

```
NCP>SET MODULE X25-PROTOCOL GROUP CUG247 NUMBER 95 DTE 16175554236
```

This command assigns group number 95 to closed user group CUG247 on DTE 16175554236.

Use the NCP SET MODULE X25/X29-SERVER commands to create or modify server module parameters in the volatile data base. Use the VNP SET MODULE X25-SERVER command to change the server module state in the system image file. (There is no VNP SET MODULE X29-SERVER command.)

Restrictions:

- The STATE parameter is valid only for SET MODULE X25-SERVER commands. It is the only valid parameter for VNP.
- You cannot modify parameters for existing destinations. You can only create new destinations. Existing destinations can be cleared, then reset with new parameters. If you specify a destination name, you must also specify the object.
- If you specify CALL MASK or CALL VALUE, you must specify both, and the *hex-value* of each must contain the same number of digits.

Format:

```
SET MODULE {X25-SERVER}  { [COUNTER TIMER seconds]
           {X29-SERVER}  { [STATE {OFF }]
                         { [      {ON  }]
                         { [      {SHUT}]
                         {
                         {  DESTINATION dest-name  OBJECT object-id
                         {                         [CALL MASK hex-value ]
                         {                         [CALL VALUE hex-value]
                         {                         [GROUP group-name]
                         {                         [NUMBER dte-address]
                         {                         [PRIORITY priority]
                         {                         [SUBADDRESSES range]
```

where

COUNTER TIMER *seconds*	Sets a timer whose expiration causes a server module counter logging event.
STATE	(Valid only for X25-SERVER commands) Sets the module's operational state.

OFF	Not available.
ON	Available for normal use.
SHUT	(Invalid for VNP) Not available for new virtual circuit connections; turns off when the last virtual circuit disconnects.

DESTINATION *dest-name*	Specifies the destination for which server parameters are to be created.
OBJECT *object-id*	Specifies the task that runs when an incoming call activates it. Use either an object name or an object type code to identify the task (see Appendix B).
CALL MASK *hex-value*	Specifies the mask applied to incoming call data before it is tested against the call value (default: no mask). Use a hexadecimal string of 2 to 32 digits (must be an even number of digits); see restrictions.
CALL VALUE *hex-value*	Specifies a string of information used to test incoming call data (default: no test string). Use a hexadecimal string of 2 to 32 digits (must be an even number of digits); see restrictions.
GROUP *group-name*	Specifies the name of a closed user group (CUG) or a bilateral closed user group (BCUG) (default: no group).
NUMBER *dte-address*	Specifies the remote DTE that originates the call for the specified destination (default: no DTE).
PRIORITY *priority*	Specifies the priority of the destination. This is used to select one of a set of destinations for which the incoming call may be valid (range: 0 to 255, where 255 is the highest priority; default: 128).
SUBADDRESSES *range*	Specifies a range of local DTE subaddresses that identifies the destination for the incoming call.

Examples:

```
NCP>SET MODULE X25-SERVER STATE SHUT
```

This command makes the module unavailable for new virtual circuit connections and turns it off when the last virtual circuit disconnects.

```
NCP>SET MODULE X29-SERVER COUNTER TIMER 600
```

This command sets the module counter timer to 600.

P

Use the NCP SET NODE CIRCUIT command to associate a DECnet circuit with a loop node name in the volatile data base. Use the VNP command to do the same with the system image file.

Format:

SET NODE *node-name* CIRCUIT *circuit-id*

where

NODE *node-name*	Specifies the loop node with which the circuit is to be associated.
CIRCUIT *circuit-id*	Specifies a DECnet circuit to be used for all traffic to the specified loop node.

Example:

```
NCP>SET NODE LASVGS CIRCUIT DMP-0.2
```

This command sets node LASVGS as the loop node name for circuit DMP-0.2.

Use the NCP SET NODE NAME command to associate a node name with a node in the volatile data base. Use the VNP command to do the same with the system image file.

Format:

SET NODE *node-id* NAME *node-name*

where

NODE *node-id*	Specifies the node for which a name is to be created or modified. The *node-id* must be a node address if you wish to assign a name to an unnamed node.
NAME *node-name*	Specifies a node name to be associated with the *node-id*.

Example:

```
NCP>SET NODE 12 NAME BLAKE
```

This command assigns the name BLAKE to node 12.

Use the NCP SET OBJECT command to create or modify object parameters in the volatile data base. Use the VNP command to do the same with the system image file.

Format:

SET OBJECT *type-code* [COPIES {*number* | SINGLE}]
[NAME *object-name*]
[USER {DEFAULT | LOGIN}]
[VERIFICATION {INSPECT | OFF | ON}]

where

OBJECT *type-code*	Specifies the object for which parameters are to be created or modified. Use an object type code from Appendix B.
COPIES *number*	Specifies the maximum number of copies of a task that can be run at once (range: 2 to 64). Use the keyword SINGLE (default) if the task is not multicopy.
NAME *object-name*	Specifies a name to be associated with the object. Use any valid RSX task name. If the task is multicopy, the name must end with $$$.
USER	Specifies the UIC under which a task is to run when the object's VERIFICATION option is set to ON or INSPECT.

DEFAULT	(Default) The task runs under the default UIC under which it was built or installed.
LOGIN	The task runs under the log-in UIC.

VERIFICATION	Specifies the degree of access to the controlled object.

INSPECT	Allows access to the object while indicating to it whether or not the access control information is valid.
OFF	(Default) Allows access to the object regardless of the validity of the access control information.
ON	Allows access to the object only for inbound connections with valid access control information.

Example:

```
NCP>SET OBJECT 19 USER LOGIN VERIFICATION ON
```

This command specifies that object 19 is to run under the log-in UIC and that access is to be allowed only with valid access control information.

Use the NCP SET PROCESS command to load processes and to modify process parameters in the volatile data base. With NCP, a network process remains in memory until you clear the system or the process (see CLEAR SYSTEM and CLEAR PROCESS) or until you reboot the system. Use the VNP SET PROCESS command to load processes and to modify process parameters in the system image file.

Restrictions:

- If you reload NW after clearing it, you must also issue the SET MODULE X25-PROTOCOL command (first with STATE OFF and then with STATE ON) to update the process pointers to the data base.
- Data space is allocated within the process's addressing space only.

Format:

```
SET PROCESS process-name  [ALL]
                          [[LOCATION] {FIRSTFIT} ]
                          [           {TOPDOWN } ]
                          [MAXIMUM CONTROLLERS count]
                          [MAXIMUM LINES count]
                          [PARTITION partition-name]
```

where

PROCESS *process-name*	Specifies the process for which parameters are to be modified. Use a process name from the table in Chapter 2.
ALL	Loads the process with all process parameter defaults specified in the permanent data base. If you specify ALL, you cannot include any other parameters.
LOCATION	Specifies the type of dynamic memory allocation for the driver process.

FIRSTFIT	(Default) Loads the process at the first available space that is large enough.
TOPDOWN	Loads the process at the top of the partition.

MAXIMUM CONTROLLERS *count*	Specifies the maximum number of hardware controllers (range: 1 to 64) for which the process is to allocate data space (see restrictions).
MAXIMUM LINES *count*	Specifies the maximum number of lines (range: 1 to 64) for which the process is to allocate data space (see restrictions).
PARTITION *partition-name*	Specifies the partition name. Use a name consisting of 1 to 6 Radix-50 characters.

Example:

```
NCP>SET PROCESS DPV MAXIMUM LINES 2
```

This command limits process DPV to the control of two lines.

P

Use the NCP or VNP SET SYSTEM command to load the Communications Executive and all network process, line, circuit, module, and node defaults that were set in the permanent data base.

Format:

SET SYSTEM [TOP]

where

TOP (Invalid for NCP) Specifies that the system be loaded at the top of memory.

Use the SET TRACE command to start (or stop) tracing the specified line(s). For more information on the trace interpreter task, see Chapter 13.

Format:

SET TRACE {**LINE** ***line-id*** | **ACTIVE LINES**} [**BUFFER SIZE** ***block-count***] [**FILE** ***file***] [**STATE** {**OFF** | **ON**}]

where

LINE *line-id*	Sets tracing for the specified line only.
ACTIVE LINES	Sets tracing for all active lines.
BUFFER SIZE *block-count*	Specifies the size (in 32-word blocks) of the trace buffer used to collect trace data (range: 1 to 255; default: 4).
FILE *file*	Specifies the name of the file to which trace data is to be copied (default: LB:[1,6]PSITRACE.SYS).
STATE	Sets the TRACE operational state to OFF or ON.

Examples:

```
NCP>SET TRACE LINE DMC-0 STATE ON BUFFER SIZE 10
```

This command turns tracing on for line DMC-0 and allocates a 10-block buffer in which to collect trace data.

```
NCP>SET TRACE ACTIVE LINES STATE OFF
```

This command turns tracing off for all active lines.

Use the NCP SHOW ALIAS command to display alias information stored in the volatile data base (see Chapter 3). Use the VNP command to do the same with the system image file.

Format:

SHOW {ALL ALIASES | ALIAS *alias-name* | KNOWN ALIASES} [{CHARACTERISTICS | SUMMARY}] [[SCOPE] *scope*] [TO *file*]

where

ALL ALIASES	Displays information for all aliases.
ALIAS *alias-name*	Displays information for the specified alias only.
KNOWN ALIASES	Displays information for all aliases for the specified scope (default scope is your terminal).
CHARACTERISTICS SUMMARY	See definitions in Section 6.5.2.
SCOPE *scope*	(Invalid for ALL ALIASES) See definition in Section 6.5.2.
TO *file*	(Valid for NCP only) See definition in Section 6.5.2.

Example:

```
NCP>SHOW ALIAS SMITH CHARACTERISTICS TERMINAL TT24: - RET
NCP>TO LB:[1,9]ALIAS.SYS
```

This command (shown in continuation format) directs static display information for alias SMITH on terminal TT24: to file LB:[1,9]ALIAS.SYS.

Use the NCP SHOW CIRCUIT command to display circuit information stored in the volatile data base (see Chapter 3). Use the VNP command to do the same with the system image file.

Format:

```
SHOW { CIRCUIT circuit-id        }  [ { CHARACTERISTICS } ]  [TO file]
     { ACTIVE CIRCUITS           }  [ { COUNTERS        } ]
     { KNOWN CIRCUITS            }  [ { STATUS          } ]
     { SIGNIFICANT CIRCUITS      }  [ { SUMMARY         } ]
```

where

CIRCUIT *circuit-id*	Displays information for the specified circuit only.
ACTIVE CIRCUITS	(Invalid for VNP) Displays information for all active circuits.
KNOWN CIRCUITS	Displays information for all known circuits.
SIGNIFICANT CIRCUITS	(Invalid for VNP) Displays information about all circuits for which information is available.
CHARACTERISTICS COUNTERS STATUS SUMMARY	See definitions in Section 6.5.2.
TO *file*	(Valid for NCP only) See definition in Section 6.5.2.

Example:

```
NCP>SHOW ACTIVE CIRCUITS COUNTERS
```

This command displays circuit error and performance statistics for all active circuits.

Use the NCP SHOW EXECUTOR command to display local node information stored in the volatile data base (see Chapter 3). Use the VNP command to do the same with the system image file.

Format:

```
SHOW EXECUTOR  [ { CHARACTERISTICS } ]  [TO file]
               [ { COUNTERS        } ]
               [ { STATUS          } ]
               [ { SUMMARY         } ]
```

where

CHARACTERISTICS COUNTERS STATUS SUMMARY	See definitions in Section 6.5.2.
TO *file*	(Valid for NCP only) See definition in Section 6.5.2.

Example:

```
NCP>SHOW EXECUTOR STATUS
```

This command displays local node dynamic information.

Use the NCP SHOW LINE command to display line information stored in the volatile data base (see Chapter 3). Use the VNP command to do the same with the system image file.

Format:

```
SHOW { LINE line-id          }  [ { CHARACTERISTICS } ]  [TO file]
     { ACTIVE LINES          }  [ { COUNTERS        } ]
     { KNOWN LINES           }  [ { STATUS          } ]
     { SIGNIFICANT LINES     }  [ { SUMMARY         } ]
```

where

LINE *line-id*	Displays information for the specified line only.
ACTIVE LINES	(Invalid for VNP) Displays information for all active lines.
KNOWN LINES	Displays information for all known lines.
SIGNIFICANT LINES	(Invalid for VNP) Displays information about all lines for which information is available.
CHARACTERISTICS COUNTERS STATUS SUMMARY	See definitions in Section 6.5.2.
TO *file*	(Valid for NCP only) See definition in Section 6.5.2.

Example:

```
NCP SHOW LINE DMC-1 SUMMARY TO LB:[1,6]LINSUM.SYS
```

This command directs the most useful information about line DMC-1 to a file named LB:[1,6]LINSUM.SYS.

Use the NCP SHOW LOGGING command to display logging information stored in the volatile data base (see Chapter 3). Use the VNP command to do the same with the system image file.

Format:

```
SHOW { ACTIVE LOGGING      }  [ { CHARACTERISTICS } ]  [TO file]
     { KNOWN LOGGING       }  [ { EVENTS          } ]  [ { KNOWN SINKS        } ]
     { LOGGING CONSOLE     }  [ { STATUS          } ]  [ { SINK NODE node-id  } ]
     { LOGGING FILE        }  [ { SUMMARY         } ]
     { LOGGING MONITOR     }
     { SIGNIFICANT LOGGING }
```

where

ACTIVE LOGGING	(Invalid for VNP) Displays all active logging information.
KNOWN LOGGING	Displays information for all known logging components.
LOGGING CONSOLE	Displays information for the logging console only.
LOGGING FILE	Displays information for the logging file only.
LOGGING MONITOR	Displays information for the logging monitor program only.
SIGNIFICANT LOGGING	(Invalid for VNP) Displays logging information about all sink types for which information is available.
CHARACTERISTICS STATUS SUMMARY	See definitions in Section 6.5.2.
EVENTS	Displays event class and type information for the given logging component.
TO *file*	(Valid for NCP only) See definition in Section 6.5.2.
KNOWN SINKS	(Default) Displays logging information for all known sink nodes.
SINK NODE *node-id*	Displays logging information for the specified sink node only.

Example:

```
NCP SHOW LOGGING FILE EVENTS SINK NODE CHI42
```

This command displays event class and type information for the logging file on node CHI42.

Use the NCP SHOW MODULE X25-ACCESS command to display remote DTE destination information stored in the volatile data base (see Chapter 3). Use the VNP command to do the same with the system image file.

Format:

SHOW MODULE X25-ACCESS {ALL DESTINATIONS | DESTINATION *dest-name* | KNOWN DESTINATIONS} [{CHARACTERISTICS | SUMMARY}]
[[SCOPE] *scope*] [TO *file*]

where

ALL DESTINATIONS	Displays information for all destinations, regardless of scope.
DESTINATION *dest-name*	Displays information for the specified destination only.
KNOWN DESTINATIONS	Displays information for all known destinations within th specified scope.
CHARACTERISTICS SUMMARY	See definitions in Section 6.5.2.
SCOPE *scope*	See definition in Section 6.5.2.
TO *file*	(Valid for NCP only) See definition in Section 6.5.2.

Example:

```
NCP SHOW MODULE X25-ACCESS ALL DESTINATIONS CHARACTERISTICS
```

This command displays static information for all destinations.

SHOW MODULE X25-PROTOCOL

Use the NCP SHOW MODULE X25-PROTOCOL command to display protocol module information stored in the volatile data base (see Chapter 3). Use the VNP command to do the same with the system image file. Some types of information can be qualified by either DTE or group.

Format:

```
SHOW MODULE X25-PROTOCOL [{DTE dte-address}]  [{CHARACTERISTICS}] [TO file]
                         [{KNOWN DTES     }]  [{COUNTERS       }]
                                              [{SUMMARY        }]

                         [{GROUP group-name}] [{CHARACTERISTICS}] [TO file]
                         [{KNOWN GROUPS    }] [{SUMMARY        }]

                         [{CHARACTERISTICS}]  [TO file]
                         [{STATUS         }]
                         [{SUMMARY        }]
```

where

DTE *dte-address*	Displays information for the specified DTE only.
KNOWN DTES	Displays information for all known DTEs.
GROUP *group-name*	Displays information for the specified group only.
KNOWN GROUPS	Displays information for all known groups.
CHARACTERISTICS COUNTERS STATUS SUMMARY	See definitions in Section 6.5.2.
TO *file*	(Valid for NCP only) See definition in Section 6.5.2.

Example:

```
NCP>SHOW MODULE X25-PROTOCOL KNOWN GROUPS SUMMARY
```

This command displays the most useful information about all known groups in the network.

SHOW MODULE X25-SERVER
SHOW MODULE X29-SERVER

Use the NCP SHOW MODULE X25/X29-SERVER commands to display server module information stored in the volatile data base (see Chapter 3). Use the VNP commands to do the same with the system image file. Some types of information can be qualified by destination.

Format:

```
SHOW  MODULE {X25-SERVER} [{DESTINATION dest-name  }] [{CHARACTERISTICS}] [TO file]
             {X29-SERVER} [{KNOWN DESTINATIONS     }] [{SUMMARY        }]

                          [{CHARACTERISTICS}]        [TO file]
                          [{COUNTERS       }]
                          [{STATUS         }]
                          [{SUMMARY        }]
```

where

DESTINATION *dest-name*	Displays information for the specified destination only.
KNOWN DESTINATIONS	Displays information for all known destinations.
CHARACTERISTICS COUNTERS STATUS SUMMARY	See definitions in Section 6.5.2.
TO *file*	(Valid for NCP only) See definition in Section 6.5.2.

Examples:

```
NCP>SHOW MODULE X25-SERVER COUNTERS TO LB:[1,6]X25CTR.SYS
```

This command directs the error and performance statistics for the module to a file named LB:[1,6]X25CTR.SYS.

```
NCP>SHOW MODULE X29-SERVER DESTINATION CHI55 CHARACTERISTICS
```

This command displays static module information for destination CHI55.

Use the NCP SHOW NODE command to display node information stored in the volatile data base (see Chapter 3). Use the VNP command to do the same with the system image file. See the SHOW EXECUTOR command for executor node displays.

Restriction:

No information will be displayed for an end node until a link has been established to it. The node may appear to be unreachable even when it is not.

Format:

```
SHOW  { NODE node-id          }   [ { CHARACTERISTICS } ]   [TO file]
      { ACTIVE NODES          }   [ { COUNTERS        } ]
      { ADJACENT NODES        }   [ { STATUS          } ]
      { KNOWN NODES           }   [ { SUMMARY         } ]
      { LOOP NODES            }
      { SIGNIFICANT NODES     }
```

where

NODE *node-id*	Displays information for the specified node only.
ACTIVE NODES	(Invalid for VNP) For a routing node, displays information for all reachable nodes. For a nonrouting node, displays information for all nodes that are (1) adjacent, (2) designated routers, or (3) connected to the executor by a logical link.
ADJACENT NODES	Displays information for all adjacent nodes.
KNOWN NODES	Displays information for all known nodes.
LOOP NODES	Displays information for all loop nodes.
SIGNIFICANT NODES	(Invalid for VNP) Displays information about all nodes for which any information is available.
CHARACTERISTICS COUNTERS STATUS SUMMARY	See definitions in Section 6.5.2.
TO *file*	(Valid for NCP only) See definition in Section 6.5.2.

Example:

```
NCP>SHOW KNOWN NODES COUNTERS
```

This command displays error and performance statistics for all known nodes.

Use the NCP SHOW OBJECT command to display object information stored in the volatile data base (see Chapter 3). Use the VNP command to do the same with the system image file.

Format:

SHOW {OBJECT *type-code* | KNOWN OBJECTS} [{CHARACTERISTICS | SUMMARY}] [TO *file*]

where

OBJECT *type-code*	Displays information for the specified object only. Use an object type code from Appendix B.
KNOWN OBJECTS	Displays information for all known objects.
CHARACTERISTICS SUMMARY	See definitions in Section 6.5.2.
TO *file*	(Valid for NCP only) See definition in Section 6.5.2.

Example:

```
NCP SHOW OBJECT 25
```

This command displays only the most useful information about object 25.

Use the NCP SHOW PROCESS command to display process information stored in the volatile data base (see Chapter 3). Use the VNP command to do the same with the system image file.

Format:

SHOW {PROCESS *process-name* | ACTIVE PROCESSES | KNOWN PROCESSES} [{STATUS | SUMMARY}] [TO *file*]

where

PROCESS *process-name*	Displays information for the specified process only. Use a process name from the table in Chapter 2.
ACTIVE PROCESSES	(Invalid for VNP) Displays information for all active processes.
KNOWN PROCESSES	Displays information for all known processes.
STATUS SUMMARY	See definitions in Section 6.5.2.
TO *file*	(Valid for NCP only) See definition in Section 6.5.2.

Example:

```
NCP>SHOW PROCESS KDP STATUS
```

This command displays dynamic information for process KDP.

NP

Use the NCP SHOW SYSTEM command to display system information stored in the volatile data base (see Chapter 3). Use the VNP command to do the same with the system image file.

Format:

SHOW SYSTEM [{ CHARACTERISTICS | COUNTERS | STATUS | SUMMARY }] [TO *file*]

where

CHARACTERISTICS See definitions in Section 6.5.2.
COUNTERS
STATUS
SUMMARY

TO *file* (Valid for NCP only) See definition in Section 6.5.2.

Example:

```
NCP SHOW SYSTEM COUNTERS
```

This command displays system error and performance statistics.

Use the SHOW TRACE command to display trace information stored in the volatile data base (see Chapters 3 and 13).

Format:

SHOW TRACE $\left[\begin{Bmatrix}\textbf{STATUS}\\ \textbf{SUMMARY}\end{Bmatrix}\right]$ **[TO *file*]**

where

STATUS SUMMARY	See definitions in Section 6.5.2.
TO *file*	See definition in Section 6.5.2.

Example:

```
NCP>SHOW TRACE TO LB:[1,6]TRACE.SYS
```

This command directs a summary (by default) of trace information to file LB:[1,6]TRACE.SYS.

Use the TELL prefix to send an NCP command to a remote node for execution. TELL sets the executor for only one command and must prefix the command for which it is intended. The TELL part of the command always executes at the local node, even though the executor may be set to a remote node identification. You can specify access control information (if required) to connect to the remote node.

Format:

TELL *node-id*[*acc-con-info*] *ncp-command*

where

node-id	Specifies the node (by name, alias, or address) to receive and execute the NCP command.
acc-con-info	Specifies access control information (if required) to connect to the node.
ncp-command	Represents any valid NCP command that is remotely executable. Appendix A lists all NCP commands and flags those that are not executable remotely.

Example:

```
NCP>TELL BOS LOAD NODE ALB
```

This command sends the LOAD NODE command to node BOS, where it executes.

Use the TRIGGER NODE command to trigger the bootstrap mechanism of a node so that the node loads itself. This command initiates the loading sequence for an unattended system. For information on down-line load operations, see Chapter 4.

Format:

TRIGGER NODE *node-id* [VIA *circuit-id*]
[PHYSICAL ADDRESS *E-address*]
[[SERVICE] PASSWORD *password*]

where

NODE *node-id*	Specifies the node whose bootstrap is to be triggered.
VIA *circuit-id*	Specifies the circuit over which the operation is to take place.
PHYSICAL ADDRESS *E-address*	(For Ethernet nodes only) Identifies the Ethernet physical address that the node currently uses to identify itself. (Required for Ethernet circuits if the hardware address parameter has not been specified in the volatile data base; see SET NODE.)
[SERVICE] PASSWORD *password*	Specifies the password required to trigger the bootstrap mechanism on the node.

NOTE

If you do not want a password echoed to your terminal while you enter it, press the RETURN key after the keyword PASSWORD. NCP prompts for the password and turns off echoing until the next prompt.

Examples:

```
NCP>TRIGGER NODE NYC
```

This command triggers the bootstrap mechanism on node NYC to initiate a down-line load.

```
NCP>TRIGGER NODE NYC VIA DMC-0 PASSWORD 5A
```

This command provides service password 5A to trigger the bootstrap mechanism on node NYC and initiate a down-line load over circuit DMC-0.

Use the TRIGGER VIA command to trigger the bootstrap mechanism of a node so that the node loads itself. This command initiates the loading sequence for an unattended system by triggering the node via the specified circuit. The node identification is obtained from the volatile data base on the executor node. For information on down-line load operations, see Chapter 4.

Format:

TRIGGER VIA *circuit-id* [PHYSICAL ADDRESS *E-address*]
[[SERVICE] PASSWORD *password*]

where

VIA *circuit-id*	Specifies the circuit over which the operation is to take place.
PHYSICAL ADDRESS *E-address*	(For Ethernet nodes only) Identifies the Ethernet physical address that the node currently uses to identify itself. (Required for Ethernet circuits if the hardware address parameter has not been specified in the volatile data base; see SET NODE.)
[SERVICE] PASSWORD *password*	Specifies the password required to trigger the bootstrap mechanism on the node.

NOTE

If you do not want a password echoed to your terminal while you enter it, press the RETURN key after the keyword PASSWORD. NCP prompts for the password and turns off echoing until the next prompt.

Examples:

```
NCP>TRIGGER VIA DMC-0
```

This command triggers the bootstrap mechanism on the node connected to circuit DMC-0.

```
NCP>TRIGGER VIA DMC-0 PASSWORD FF
```

This command provides service password FF to trigger the bootstrap mechanism on the node connected to circuit DMC-0.

Use the ZERO CIRCUIT command to zero circuit counters for the specified circuit(s). The executor node maintains these counters on a per circuit basis.

Format:

ZERO $\left\{\begin{array}{l}\text{CIRCUIT } \textit{circuit-id}\\ \text{KNOWN CIRCUITS}\end{array}\right\}$ [COUNTERS]

where

CIRCUIT *circuit-id*	Zeroes counters for the specified circuit only.
KNOWN CIRCUITS	Zeroes counters for all known circuits.

Example:

```
NCP>ZERO CIRCUIT DMC-0
```

This command zeroes circuit counters for circuit DMC-0.

Use the ZERO EXECUTOR command to zero node counters associated with and maintained on the executor node.

Format:

ZERO EXECUTOR [COUNTERS]

Use the ZERO LINE command to zero line counters for the specified line(s). The executor node maintains these counters on a per line basis.

Format:

ZERO {LINE *line-id* / KNOWN LINES} [COUNTERS]

where

LINE *line-id*	Zeroes counters for the specified line only.
KNOWN LINES	Zeroes counters for all known lines.

Example:

```
NCP>ZERO LINE DUV-3
```

This command zeroes line counters for line DUV-3.

Use the ZERO MODULE X25-PROTOCOL command to zero module counters for the specified DTE(s). The executor node maintains these counters on a per DTE basis.

Format:

ZERO MODULE X25-PROTOCOL {DTE *dte-address* | KNOWN DTES} [COUNTERS]

where

DTE *dte-address*	Zeroes counters for the specified DTE only.
KNOWN DTES	Zeroes counters for all known DTEs.

Example:

```
NCP>ZERO MODULE X25-PROTOCOL DTE 123456789
```

This command zeroes X.25 protocol module counters for DTE 123456789.

ZERO MODULE X25-SERVER ZERO MODULE X29-SERVER

Use the ZERO MODULE X25/X29-SERVER commands to zero X.25/X.29 server module counters that are maintained on the executor node.

Format:

ZERO MODULE $\left\{\begin{array}{l}\textbf{X25-SERVER}\\ \textbf{X29-SERVER}\end{array}\right\}$ **[COUNTERS]**

P

Use the ZERO NODE command to zero node counters for the specified node(s). The executor node maintains node counters on a per node basis.

Format:

$$\text{ZERO} \begin{Bmatrix} \text{NODE } \textit{node-id} \\ \text{KNOWN NODES} \end{Bmatrix} \text{ [COUNTERS]}$$

where

NODE *node-id*	Zeroes counters for the specified node only.
KNOWN NODES	Zeroes counters for all known nodes.

Example:

```
NCP>ZERO NODE ATL
```

This command zeroes node counters for node ATL.

Use the ZERO SYSTEM command to zero system counters associated with and maintained on the executor node.

Format:

ZERO SYSTEM [COUNTERS]

6.5.4 RSX-11S NCP Command Summary

This section summarizes the NCP and NICE commands that are supported by RSX-11S. This is a subset of the commands supported on RSX-11M and RSX-11M-PLUS operating systems. Unless otherwise noted in the restriction section of an individual command description, all of the following commands can be initiated both locally (that is, at the RSX-11S node) and remotely (not at the RSX-11S node).

The commands are presented in alphabetical order, using the same definitions described in Section 6.5.2. To distinguish this subset from the full set, NCP/S is printed at the top of each page above the command name. In addition, each command is designated as privileged (P) or nonprivileged (NP).

Use the LOOP NODE command to test a node in the network. You can include access control information if the node requires it. If you are testing the executor node, you can use the LOOP EXECUTOR command. Either command causes test blocks of data to be transmitted to the specified node. See Chapter 3 for more information on loopback testing.

Format:

```
LOOP {NODE node-id[acc-con-info]}  [COUNT count]
     {EXECUTOR                  }  [LENGTH length]
                                   [WITH {MIXED }]
                                   [     {ONES  }]
                                   [     {ZEROES}]
```

where

NODE *node-id*	Specifies a node for loopback testing.
acc-con-info	Specifies access control information, if required.
EXECUTOR	Specifies the executor node for loopback testing.
COUNT *count*	Specifies the number of blocks to be sent during loopback testing (range: 1 to 65535; default: 1 block is looped).
LENGTH *length*	Specifies the length (in bytes) of blocks to be sent during loopback testing. When testing over the Ethernet, the length must be a decimal integer in the range of 32 to 1484. Otherwise, the length must be a decimal integer in the range of 1 to *n*, where *n* must be less than the smaller of either the local looper buffer size or the remote mirror buffer size (default: 40 bytes).
WITH	Specifies the type of binary information to be sent during testing (default: MIXED — a combination of ones and zeroes).

Example:

```
NCP LOOP NODE NYC COUNT 10
```

This command loops 10 blocks of mixed test messages to remote node NYC. Each block is 40 bytes.

Use the SET CIRCUIT command to set the operational state for the specified circuit in the volatile data base.

Format:

SET CIRCUIT *circuit-id* STATE $\left\{\begin{array}{l}\text{OFF}\\ \text{ON}\end{array}\right\}$

where

CIRCUIT *circuit-id*	Specifies the circuit whose state is to be set.
STATE	Sets the circuit's operational state to ON or OFF.

Example:

```
NCP>SET CIRCUIT KDZ-1-3 STATE OFF
```

This command sets the state of circuit KDZ-1-3 to OFF.

SET EXECUTOR HOST

Use the SET EXECUTOR HOST command to specify a host node in the volatile data base.

Format:

SET EXECUTOR HOST *node-address*

where

HOST *node-address*	Specifies the address for the host node to be used for down-line load, trigger, and up-line dump service operations.

Example:

```
NCP>SET EXECUTOR HOST 08
```

This command sets node 08 as host node for down-line load, trigger, and up-line dump.

P

Use the SET LINE command to set the controller mode for a specified loaded line in the volatile data base.

Restrictions:

- SET LINE cannot be initiated from a remote node.
- The specified line must be in the ON or OFF state (that is, not CLEARED).

Format:

SET LINE *line-id* CONTROLLER $\left\{\begin{array}{l}\text{LOOPBACK}\\ \text{NORMAL}\end{array}\right\}$

where

LINE *line-id*	Specifies the line for which the controller mode is to be specified.
CONTROLLER	(For Ethernet, DMC/DMR, DMP, and DMV lines) Specifies the controller mode for the line. The line's circuit must be in the OFF state.
	LOOPBACK — Internal device loopback mode
	NORMAL — Normal operating mode

Example:

```
NCP>SET LINE DMC-0 CONTROLLER LOOPBACK
```

This command sets the operating mode for line DMC-0 to loopback mode.

SET LOGGING CONSOLE

Use the SET LOGGING CONSOLE command to modify the operational state of the logging console in the volatile data base.

Format:

SET LOGGING CONSOLE STATE $\begin{Bmatrix} \text{OFF} \\ \text{ON} \end{Bmatrix}$

where

STATE — Sets the operational state of the logging console on the executor node. When the state is OFF, events are discarded.

Example:

```
NCP>SET LOGGING CONSOLE STATE OFF
```

This command sets the state of the logging console to OFF.

Use the SHOW CIRCUIT command to display circuit information stored in the volatile data base (see Chapter 3).

Restriction:

ACTIVE CIRCUITS and KNOWN CIRCUITS cannot be initiated locally.

Format:

$$\text{SHOW} \begin{Bmatrix} \text{CIRCUIT } \textit{circuit-id} \\ \text{ACTIVE CIRCUITS} \\ \text{KNOWN CIRCUITS} \end{Bmatrix} \left[\begin{Bmatrix} \text{COUNTERS} \\ \text{STATUS} \\ \text{SUMMARY} \end{Bmatrix} \right]$$

where

CIRCUIT *circuit-id*	Displays information for the specified circuit only.
ACTIVE CIRCUITS	Displays information for all active circuits.
KNOWN CIRCUITS	Displays information for all known circuits.
COUNTERS STATUS SUMMARY	See definitions in Section 6.5.2.

Example:

```
NCP>SHOW CIRCUIT DMC-0 COUNTERS
```

This command displays error and performance statistics for circuit DMC-0.

Use the SHOW EXECUTOR command to display local node information stored in the volatile data base (see Chapter 3).

Format:

```
SHOW EXECUTOR  [ { CHARACTERISTICS } ]
               [ { COUNTERS        } ]
               [ { STATUS          } ]
               [ { SUMMARY         } ]
```

where

CHARACTERISTICS See definitions in Section 6.5.2.
COUNTERS
STATUS
SUMMARY

Example:

```
NCP>SHOW EXECUTOR STATUS
```

This command displays local node dynamic information.

Use the SHOW LINE command to display line information stored in the volatile data base (see Chapter 3).

Restriction:

ACTIVE LINES and KNOWN LINES cannot be initiated locally.

Format:

$$\text{SHOW} \begin{Bmatrix} \text{LINE } \textit{line-id} \\ \text{ACTIVE LINES} \\ \text{KNOWN LINES} \end{Bmatrix} \left[\begin{Bmatrix} \text{COUNTERS} \\ \text{STATUS} \\ \text{SUMMARY} \end{Bmatrix} \right]$$

where

LINE *line-id*	Displays information for the specified line only.
ACTIVE LINES	Displays information for all active lines.
KNOWN LINES	Displays information for all known lines.
COUNTERS STATUS SUMMARY	See definitions in Section 6.5.2.

Example:

```
NCP>SHOW LINE DMC-1 SUMMARY
```

This command displays the most useful information about line DMC-1.

Use the SHOW LOGGING CONSOLE command to display dynamic logging console information stored in the volatile data base (see Chapter 3).

Format:

SHOW LOGGING CONSOLE STATUS

Use the SHOW NODE command to display node information stored in the volatile data base (see Chapter 3). See the SHOW EXECUTOR command for executor node displays.

Restrictions:

- ACTIVE NODES and KNOWN NODES cannot be initiated locally.
- No information will be displayed for an end node until a link has been established to it. The node may appear to be unreachable even when it is not.

Format:

```
SHOW  { NODE node-id  }   [ { CHARACTERISTICS } ]
      { ACTIVE NODES  }   [ { COUNTERS        } ]
      { KNOWN NODES   }   [ { STATUS          } ]
                          [ { SUMMARY         } ]
```

where

NODE *node-id*	Displays information for the specified node only.
ACTIVE NODES	Displays information for all active nodes.
KNOWN NODES	Displays information for all known nodes.
CHARACTERISTICS COUNTERS STATUS SUMMARY	See definitions in Section 6.5.2.

Example:

```
NCP>SHOW NODE LOSANG COUNTERS
```

This command displays error and performance statistics for node LOSANG.

Use the SHOW SYSTEM command to display system information stored in the volatile data base (see Chapter 3).

Format:

```
SHOW SYSTEM [ { CHARACTERISTICS } ]
            [ { COUNTERS        } ]
            [ { STATUS          } ]
            [ { SUMMARY         } ]
```

where

CHARACTERISTICS COUNTERS STATUS SUMMARY	See definitions in Section 6.5.2.

Example:

```
NCP>SHOW SYSTEM COUNTERS
```

This command displays system error and performance statistics.

Use the ZERO CIRCUIT command to zero circuit counters for the specified circuit(s). The executor node maintains these counters on a per circuit basis.

Restriction:

KNOWN CIRCUITS cannot be initiated locally.

Format:

$$\text{ZERO} \begin{Bmatrix} \text{CIRCUIT } \textit{circuit-id} \\ \text{KNOWN CIRCUITS} \end{Bmatrix} \text{[COUNTERS]}$$

where

CIRCUIT *circuit-id*	Zeroes counters for the specified circuit only.
KNOWN CIRCUITS	Zeroes counters for all known circuits.

Example:

```
NCP>ZERO CIRCUIT DUP-0
```

This command zeroes circuit counters for circuit DUP-0.

ZERO EXECUTOR

Use the ZERO EXECUTOR command to zero node counters associated with and maintained on the executor node.

Format:

ZERO EXECUTOR [COUNTERS]

Use the ZERO LINE command to zero line counters for the specified line(s). The executor node maintains these counters on a per line basis.

Restriction:

KNOWN LINES cannot be initiated locally.

Format:

ZERO {LINE *line-id* | KNOWN LINES} [COUNTERS]

where

KNOWN LINES	Zeroes counters for all known lines.
LINE *line-id*	Zeroes counters for the specified line only.

Example:

```
NCP>ZERO LINE DUV-3
```

This command zeroes line counters for line DUV-3.

Use the ZERO NODE command to zero node counters for the specified node(s). The executor node maintains node counters on a per node basis.

Restriction:

This command cannot be initiated locally.

Format:

$$\text{ZERO} \left\{ \begin{array}{l} \text{NODE } \textit{node-id} \\ \text{KNOWN NODES} \end{array} \right\} \text{[COUNTERS]}$$

where

NODE *node-id*	Zeroes counters for the specified node only.
KNOWN NODES	Zeroes counters for all known nodes.

Example:

```
NCP>TELL NET11S ZERO KNOWN NODES
```

This command zeroes node counters for all known nodes.

Use the ZERO SYSTEM command to zero system counters associated with and maintained on the executor node.

Format:

ZERO SYSTEM [COUNTERS]

PART III
NETWORK MANAGEMENT TOOLS

Each of the chapters in Part III describes the operation of a network management tool, including commands for its operation and examples of its use and/or output (where applicable).

- Chapter 7 — Console Carrier Requester (CCR)
- Chapter 8 — Host Task Loader (HLD)
- Chapter 9 — KMX Dump Analyzer
- Chapter 10 — Network Dump Analyzer (NDA)
- Chapter 11 — Network Display Program (NTD)
- Chapter 12 — Queue Manager (QUE)
- Chapter 13 — Trace Interpreter Task (TRI)

Chapter 7
Console Carrier Requester (CCR)

The console carrier requester (CCR) uses the DLX interface to communicate with the console carrier server (CCS) residing on a target node to provide remote access to normal console services. CCR is used to emulate the console on a remote Digital Ethernet Communication Server (DECS) system.

7.1 Running CCR

The privileged user who wishes to invoke CCR and communicate with the console carrier server must have the PLUTOWL.SYS and PLUTOCC.SYS files under the network UIC. To access CCR (which has an installed task name of ...CCR), use the following command:

CCR NODE *node-id* [SERVICE CIRCUIT *circuit-id*]
[SERVICE PASSWORD *password*]
[PHYSICAL ADDRESS *E-address*]

where

NODE *node-id*	specifies the target node by address (range: 1 to 1023) or name (1 to 6 alphanumeric characters, including at least 1 alphabetic character).
SERVICE CIRCUIT *circuit-id*	identifies the circuit to the target node. (*circuit-id* has the format *dev-c*[-*u*][.*t*].)
SERVICE PASSWORD *password*	defines the password required to access the target node. The password is a hexadecimal number in the range of 0 to FFFFFFFFFFFFFFFF (up to 16 hexadecimal digits).
PHYSICAL ADDRESS *E-address*	identifies the Ethernet physical address that the node currently uses to identify itself. (Required if the hardware address parameter has not been specified in the volatile data base; see the SET NODE command in Chapter 6.)

The *node-id* is required. If the other parameters are not specified in the command line, they must be specified in the down-line load data base. If they are specified in the command line, they override the parameters set in the down-line load data base.

7.2 Reserving the Console

The console carrier server can be in one of three states when you wish to reserve the console:

- Loaded and unreserved
- Loaded and reserved
- Not loaded

If the console carrier server is loaded and unreserved, the console carrier requester reserves it, and the following message is displayed on your terminal:

```
CCR -- Remote console reserved
```

If the console carrier server is loaded and reserved, an error message is displayed on your terminal. (A complete list of CCR error messages is given in Chapter 14.)

If the console carrier server is not loaded, the requester loads the server. It may be necessary for the requester to identify the target node via the 12-hexadecimal-digit Ethernet physical address if the target node is not known to the network. Once the server is loaded, the requester attempts to reserve the console and proceed as described above.

7.3 CCR Special Characters

Two special characters are supplied with the CCR software:

CTRL/B Operates as a BREAK command to get the attention of the console on-line debugging tool (ODT).

CTRL/D Initiates an exit from console carrier mode.

7.4 Sample CCR Session

In the following sample session, the user invokes CCR to test node DALLAS.

```
>CCR NODE DALLAS
CCR -- Remote console is reserved
CTRL/B
```

The system then displays the current program counter and a prompt (@). At this point, you can enter commands as if you were physically located at a console. The command is transmitted by the requester to the server, and the appropriate response is transmitted by the server to the requester. The requester then displays this information.

To examine a register, enter $ followed by the register number and a slash (/). To examine a location, enter the address of the location and a slash. The system then displays the contents on the same line. Enter LF to view the next register or location, or enter RET to return to the @ prompt. When you wish to exit ODT, enter P (proceed). You must then issue a CTRL/D to return to the RSX system prompt.

```
002160
@$0/00000 LF
R1/005621LF
R2/112000LF
R3/000000LF
R4/000001LF
R5/000000RET
@100/016122LF
000102/000340LF
000104/001614LF
000106/000341RET
@PRET
CTRL D CCR -- Remote console is released
>
```

Chapter 8
Host Task Loader (HLD)

The Host Task Loader (HLD) utility is the down-line task loader service task that runs on the host system. HLD communicates with the satellite task loader (SLD) running on a remote RSX-11S system to enable tasks on the host system to be down-line loaded to the remote node. You build SLD during RSX-11S network generation, and you build HLD during RSX-11M/M-PLUS network generation. For more information, see the *DECnet-RSX Network Generation and Installation Guide*.

HLD is driven by an external mapping table that maps requests from a task's SLD to a file residing on the host system. This chapter describes how to build and use this mapping table. For more information on down-line task loading, see Chapter 4.

8.1 Formatting the HLD Mapping Table

The mapping table is divided into two logical sections, each containing a different type of task:

General purpose tasks are listed first in the table. They are not associated with any particular node and can be down-line loaded to any RSX-11S node in the network. Such tasks cannot be checkpointed. Guidelines for specifying and using general purpose tasks follow:

- You can specify the same task name more than once in the general purpose task list. This allows RSX-11S systems to share installed task names.
- Task mapping is automatically performed unless you specify otherwise.
- LUN fixing (see Section 8.3) is automatically performed.
- You must install general purpose tasks in the same partition for all RSX-11S systems, although the partition addresses need not be identical for mapped systems.
- If you add new nodes to the network, existing general purpose tasks can be down-line loaded to those nodes without your having to change the mapping table. Nodes receiving only general purpose tasks do not have to be defined in the HLD mapping table.

Node-specific tasks are always preceded by a node name in the table and can be down-line loaded to that node only. If you wish LUN fixing to be performed, you must so specify (see Section 8.3).

8.2 Creating/Modifying an HLD Mapping Table

To create or modify an external HLD mapping table image, execute the [x,1]HLDDAT.CMD command file provided by NETGEN (x = group code specified during NETGEN). This command file leads you through the steps required to specify your tasks. To get help at any time during the procedure, enter ESC in response to a prompt.

The first prompt is for a command string. You can enter any of the following commands:

DEFINE	Creates a new node and/or task entries.
EXIT	Terminates the session, replacing the old data base contents and optionally assembling and building the new data base (same as CTRL/Z).
HELP	Displays a list of the HLD commands.
LIST	Displays all node and task entries.
PURGE	Deletes all old node and/or task entries.
QUIT	Terminates the session with no changes to the old data base contents.

First you define any general purpose tasks that you wish to create or modify. Enter the task image file name in response to the prompt. You are then prompted to enter a task name if you wish it to be different from the task image file name. You can also override task mapping if you wish. When you have made all your general purpose task entries, type RET to begin node-specific task definitions.

If there are any nodes already specified, the system displays each node name in turn and asks if you want to define any new tasks for it. For each task that you define, you can choose whether or not you want LUN fixing performed (see Section 8.3). Press RET when you are ready to define tasks for a different node. The system will either display the name of another node that has already been specified, or it will ask if you would like to specify a new node.

When you have made all your task specifications, enter EXIT. The system will then ask if you want to build the new HLD data base now. If you do, it prompts for the DECnet object kit device (*ddu*, where *dd* is the device and *u* is the unit number). If you choose to build the data base later, you can either execute the HLDDAT command file again and issue EXIT, or you can mount the DECnet object kit on disk device *ddu* and issue the following commands to create the task image file.

```
ASN ddu:=IN:
SET /UIC=[100,24]
MAC @HLDTABASM
TKB @HLDTABBLD
```

8.3 LUN Fixing

LUN fixing is an SLD feature that reinitializes the LUNs after the task has been down-line loaded. (This is equivalent to the user program issuing ALUN$ assignments at the start of the program.) This feature allows a single task to be loaded into multiple RSX-11S systems that may have different unit control block addresses.

LUN fixing causes the task transfer to be followed by a copy of the ASCII device assignments. SLD then assigns the loading task's LUNs dynamically.

Tasks to be down-line loaded must be installed into the RSX-11S system. This initializes the task's LUN assignments and prepares the task image file for loading. It also creates the RSX-11S system data bases that allow the tasks to be run.

LUN fixing is automatic for general purpose tasks. You must specify it if you wish to perform LUN fixing on a node-specific task.

8.4 HLD Error Handling

The HLD utility returns error messages resulting from file access and network communication operations and from inconsistencies in the mapping table and file to be loaded. Chapter 14 provides a complete list of HLD error messages and their meanings.

Chapter 9
KMX Dump Analyzer (KDA)

If the performance of a KMX device deteriorates and this cannot be attributed to a hardware fault, you can dump the KMX microcode for analysis by your local software specialist. The NCP KMX-DUMP command controls the dumping of the microcode to disk (see Chapter 6). The KMX dump analyzer task (KDA) formats and prints the contents of the KMX dump disk file.

You must have included both the dump KMX task (DUK) and the KMX dump analyzer task (KDA) in your system during NETGEN if you want to use this facility. The *RSX-11 PSI Generation Guide* gives details of the DUK and KDA tasks.

9.1 Invoking and Exiting KDA

Use any RSX-11 method of invoking tasks to enter KDA. The simplest method is to type:

```
MCR>KDA
```

Once KDA is running, it returns the prompt:

```
KDA>
```

Exit from KDA by pressing CTRL/Z.

9.2 Using the KDA Command

The KDA command has the following format:

[*outfile*[*switches*]=][*infile*[*switches*]]

where

outfile — is the file specification for the analyzed dump listing file (default is SY:[UFD]*inputname*.LST, where UFD is the user's default user file directory). The = symbol marks this as the output file specification.

infile — is the file specification for the binary input file containing the dump (default is LB:[1,6]PSIKMXMEM.SYS). Specify the same file that you specified for the output file in the KMX-DUMP LINE command.

switches — are options to control the listing that KDA produces. Any compatible combination of the switches shown in Table 9-1 can be specified for either the input file or the output file.

Table 9–1: KDA Command Switches

Switch	Meaning	Comments
/ALL	Analyze and print the control memory	
/-ALL	Analyze and print only the data memory and registers	Default switch
/SP	Spool the output file	Default switch (Output is spooled unless /-SP is specified.)
/-SP	Do not spool the output file	Output file is retained for printing later

Example:

```
>KDA
KDA>VORTEX/ALL=DK0:[7,21]FROGST.DMP
```

This command uses input file FROGST.DMP from UFD [7,21] on DK0: and prints an analysis of control memory to output file SY:[UFD]VORTEX.LST.

Chapter 10
Network Dump Analyzer (NDA)

The Network Dump Analyzer (NDA) utility helps you to determine the cause of network crashes. NDA is a nonprivileged task for use only with DECnet-RSX systems, including those with a PSN capability. If you have selected crash dump analysis support during RSX system generation, then you can also select network crash dump analysis support during DECnet-RSX network generation. Refer to the *RSX-11M System Generation and Installation Guide* for information on obtaining crash dump support for your operating system. Refer to the *DECnet-RSX Network Generation and Installation Guide* for information on obtaining network crash dump support for DECnet-RSX.

NDA performs crash dump analysis for the network data structures in the same way that the RSX crash dump analyzer (CDA) analyzes the operating system data structures. Specifically, NDA compiles and analyzes information from the system's executive symbol table file, the network symbol table file, and the crashed system image. The information is then output in a listing file or at a line printer in a format that is useful to you. Note that an effective use of NDA requires an extensive knowledge of DECnet-RSX internal data structures.

Command switches provide you with the means to control NDA operation. There are two types of switches: analysis control switches and function control switches. Analysis control switches determine which NDA analysis routines are applied to the input file for analysis. These switches provide such options as:

- Dumping the contents of a specified octal address range of memory
- Listing general system data (time of crash, registers in use, and so forth)
- Listing the routing parameters and data bases
- Listing the remote node names and alias node names
- Listing PSI data structures (local and remote DTE descriptor blocks, NW: data base and window blocks, and so forth)
- Listing the network services protocol (ECL) data bases

Function control switches provide a number of NDA control options, such as:

- Identifying the starting block number of the secondary storage device
- Terminating the analysis after a specified number of errors has been encountered
- Limiting the number of pages of analysis output
- Creating a save file and transferring a specified amount of the crash dump to it

You can type NDA commands at a terminal, or you can place them in an indirect command file.

10.1 NDA Operation

NDA operates in the following manner:

1. First, it reads the contents of the memory dump created by the executive crash dump routine.
2. Next, it analyzes the dump in accordance with information contained in a combined executive/network symbol table file.
3. Finally, it formats and outputs the dump to a line printer or a listing file for your evaluation.

10.2 System Requirements

To obtain a crash dump that can be handled by NDA, you must first build the executive crash dump routine into your system during phase 1 of an RSX-11 SYSGEN. This is accomplished by entering B in response to the query:

```
50. Crash dump analysis: A- PANIC B- Crash dump analysis (CDA)
```

If you select this option, code is included in your system to dump memory to a secondary storage device following a system crash. Your secondary storage device can be any one of a number of different types of disk units, a magtape unit, or a DECtape unit. You can find complete information on RSX-11 SYSGEN in the *RSX-11M System Generation and Installation Guide.*

10.3 Obtaining a Crash Dump

To obtain a crash dump, control of the processor must be transferred to the executive crash dump routine following a system crash. How this control is transferred depends on the way in which the crash occurred and whether or not you have built the executive debugging tool (XDT) into your system at SYSGEN.

System crashes occur in three ways:

- The processor encounters an unintentional halt instruction (000000).
- An infinite loop condition occurs.
- The processor encounters a program condition causing it to trap.

When a system crash is the result of an unintentional halt instruction or an infinite loop condition, you must restart your processor at location 40.

When a program condition causes a processor trap and XDT is included in your system, control transfers automatically to XDT. When you type X at your terminal, XDT transfers control to the executive crash dump routine. When XDT is not included in your system, a processor trap causes control to be transferred to the crash dump routine. Refer to the *RSX-11M Guide to Writing an I/O Driver* for a description of XDT.

Regardless of the manner in which control is transferred, once the executive crash dump routine is entered, it prints the following message on the crash notification device specified during SYSGEN:

```
CRASH - CONT WITH SCRATCH MEDIA ON xxx
```

where *xxx* is the mnemonic for the device that is to receive the crashed system image.

You can then put the secondary crash dump device on line and depress the CONT switch on the CPU console. The executive crash dump routine dumps memory to the crash dump device and halts the processor upon completion.

10.4 Running NDA

You can run NDA as either an installed or an uninstalled task.

10.4.1 Running NDA as an Installed Task

When you run NDA as an installed task, the command format that you use determines where control is returned upon completion of the analysis:

- To return control to the executive:

  >NDA *command string*

- To return control to NDA:

  >NDA
  NDA> *command string*

10.4.2 Running NDA as an Uninstalled Task

When you run NDA as an uninstalled task, control is always returned to NDA upon completion of the analysis. The commands for running NDA are:

RUN $NDA NDA>*command string*	This requires NDA.TSK to be present in the user file directory corresponding to the system UIC on device LB:.
RUN NDA NDA>*command string*	This requires NDA.TSK to be present in the user file directory corresponding to the current UIC on device SY:.

10.5 NDA Command Syntax

Use the following format to input commands to NDA:

[*analysis*[/*sw1*]],[*save*[/MEMSIZ:*n*]]=[*symbols*/STB], [*input*[/*sw2*]]

where

analysis — is the crash dump analysis listing file. If a file extension is not specified, the default extension is .LST. If an analysis file is not specified, no analysis list file is created.

/sw1 — consists of the /EXIT:*n*, /LIMIT:*n*, and /-SP functional control switches for the analysis file. A description of each switch and its default value are given in Table 10-3.

save — is the crash dump save file. This file is a copy of the binary data that was written to the dump device by the dump routine. It allows you to create a historical record of crash dumps.

/MEMSIZ:*n*	is the number of K words of information to be transferred to the save file. A description of the MEMSIZ:*n* functional control switch and its default value are given in Table 10-3.
symbols/STB	is the symbol table file for the crashed system. This file must define all executive and network symbols. A description of /STB and its default value are given in Table 10-3.
input	is the binary input to NDA. This specification can be either a device name (your secondary storage device) or the name of a crash dump save file.
/*sw2*	consists of all the analysis control switches and three functional control switches — /KMR, /SYM, and /BL:*n*. A description of the analysis control switches is provided in Section 10.7.1. A description of the /KMR, /SYM, and /BL:*n* switches and their default values are given in Table 10-3.

Output specifications (to the left of the equal sign) are position dependent. Therefore, when you include both output specifications (analysis, save), you must place them in the command string in the positions shown. If you do not enter a specification for the analysis file, you must place a comma before the save file specification. The comma can appear only if a save file is specified.

Input specifications (to the right of the equal sign) are position independent and can appear in either order.

Any of the specifications in the command string can appear in the complete FILES-11 format — that is, complete with device, UIC, file name, file type, and version number. When you omit any of the command specifications, NDA uses the defaults shown in Table 10-1.

Table 10-1: NDA File Default Values

File	Device	User File Directory (UFD)	Default File Name	Value Type
Analysis file	SY:	Current	None	.LST
Save file	SY:	Current	None	.CDA
Symbols file	SY:	Current	CEX	.STB
Input file	SY:	Current	None	.CDA

The following examples illustrate the NDA file default values. Assume that the user in these examples is logged in under UIC [301,356] and that the secondary storage device is DK1:. Also assume that NDA is being operated as an installed task.

Example 1:

```
>NDA
NDA>18APR82,18APR82=CEX.STB/STB,DK1:/ALL
```

This command string creates an analysis file, 18APR82.LST, and a save file, 18APR82.CDA. The binary input is read from RK05 disk unit 1. NDA analyzes it in accordance with the information contained in the symbol definition file CEX.STB under UFD [301,356].

Example 2:

```
>NDA
NDA>,13JAN82=DK1:
```

This command string creates a save file, 13JAN82.CDA. In this case, NDA performs no analysis, since there is no analysis listing file. The binary input is read from RK05 disk unit 1.

Example 3:

```
>NDA
NDA>13JAN82=13JAN82/ALL
```

This command string produces an analysis output listing using the /ALL analysis switch (see Table 10-2 for a complete description of this switch). The binary input is read from the previously created save file (13JAN82.CDA). NDA analyzes it in accordance with the information contained in the symbol definition file CEX.STB under UFD [301,356].

Example 4:

```
>NDA
NDA>13JAN82,13JAN82/MEMSIZ:512=DK1:/ALL
```

This command string creates an analysis file named 13JAN82.LST and a save file named 13JAN82.CDA. The number of words of information to be transferred is 512K words. The input file is located on RK05 disk unit 1. NDA produces an analysis output listing using the /ALL analysis switch.

10.6 Indirect Command Files

You can enter command strings to NDA directly from the keyboard or indirectly through the use of the indirect command file facility. NDA indirect command files must not contain a reference to another command file.

10.7 NDA Switches

Switches provide you with the means to control NDA operation. There are two types: analysis control switches and functional control switches.

Analysis control switches determine which NDA analysis routines are to be applied to the input file. Some switches apply only to the DECnet software, while others apply only to the PSI software running with DECnet on a DECnet-RSX system with a PSN capability.

Functional control switches provide a number of NDA control options. For example, such switches might terminate an analysis after NDA has encountered a specified number of errors. Or, a switch might limit the number of pages of output listing.

10.7.1 Analysis Control Switches

Table 10-2 provides a list of the analysis control switches and a brief description of their use. More than one analysis control switch can be specified. If none are specified, the default is /SYS/CEX.

Table 10–2: NDA Analysis Control Switches

Switch	Description
/ALL	Executes the following switches (each is defined later in this table): /CEX /LDTE /OBJ /PRO /RDTE /SYS /CTB /LIS /PDV /PSV /REM /TSK /CUG /NSP /PHB /PVC /ROU /XACP /DST /NW /PLI /QUE /SLT DECnet, PSI, and X.25 Gateway components are analyzed only if the respective software was in the system that was dumped. /ALL is the recommended switch to use. If problems occur, then try using specific switches.
/CEX	Lists general Communications Executive data. This includes the node name and number, the current network process, and the current state of buffer usage.
/CTB	Prints information on all controller blocks assigned to network devices. (This is printed only on RSX-11M-PLUS systems.)
/CUG	Prints the PSI/X.25 Gateway closed user group name blocks.
/DST	Prints the PSI/X.25 Gateway X.25 and X.29 destination blocks.
/DMP:*a*:*b* /DUMP:*a*:*b*	Lists the contents of physical memory from octal address *a* through octal address *b*. Both *a* and *b* are 32-bit addresses.
/LDTE	Prints the PSI/X.25 Gateway local DTE descriptor blocks.
/LIS /LIST /LISTS /FREE	Scans the CCB, SDB, and RDB free lists for errors and prints the results.
/NSP	Prints the DECnet ECL data bases, including the logical link blocks.
/NW	Prints the PSI NW: data base and window blocks.
/OBJ /OBJECT	Lists the network task object types.
/PDV	Prints interpreted contents of the process description vectors.
/PHB	Prints the contents of the PSI/X.25 Gateway PLI home block.
/PLI	Prints the PSI PLI/X.25 Gateway data base and the X.25 circuit blocks.
/PRO /PROC	Dumps the line tables for all loaded network processes.
/PSV	Prints the X.25 Gateway PSV process data bases, circuit data bases, and buffer chains.
/PVC	Prints the PSI/X.25 Gateway permanent virtual circuit (PVC) name blocks.
/QUE /QUEUE	Lists the contents of the fork process queue.
/RDTE	Prints the PSI remote DTE descriptor blocks.

Table 10–2 (cont.): NDA Analysis Control Switches

Switch	Description
/REM /REMOTE	Lists the DECnet remote node names and alias names.
/ROU /ROUT /ROUTING	Lists the DECnet routing parameters, including the physical link blocks.
/SLT	Prints interpreted contents of the system line tables.
/SYS	Lists general system data. This includes the time of the crash, the current task, and the registers in use.
/TSK /TASK	Prints the network mailbox, window blocks, and I/O packets for each task using the network.
/XACP	Prints the CCBs and I/O packets waiting to be processed by the PSI X25ACP.

10.7.2 Function Control Switches

Table 10–3 provides a list of the function control switches and a brief description of their use. More than one function control switch can be specified. If switches are not specified, the default values listed in Table 10–3 are assumed. Note that when a switch requires a numeric argument, the argument is considered to be octal unless followed by a decimal point.

10.8 NDA Error Messages

NDA prints an error message on your terminal when it detects an error condition. NDA error messages are summarized in Chapter 14. Do not confuse these messages with the diagnostic analysis messages that NDA generates during the analysis and prints in the analysis listing.

Table 10-3: NDA Function Control Switches

Switch	File	Description	Default
Default			
/BL:*n*	Input file	Identifies the starting block number of the secondary storage device. The value of *n* must be less than 65535. NDA reads the dump from the input device beginning at logical block *n*. If the crash dump device is not a disk or DECtape, NDA ignores this switch.	/BL:1
/EXIT:*n*	Analysis file	Terminates analysis after encountering *n* analysis errors. NDA maintains an error count. If you specify the /EXIT:*n* switch, NDA terminates analysis after *n* errors. If you specify /EXIT but do not specify *n*, NDA exits after one error.	/-EXIT
/KMR	Input file	When NDA reads incorrect page address register (PAR) values from the crash stack, it prints an error message on the user's terminal and attempts to continue with the analysis. In this event, you can use the /KMR switch when you restart the analysis. /KMR forces NDA to use standard mapping values to convert kernel virtual addresses to physical memory addresses.	
/LIMIT:*n*	Analysis file	Limits the number of pages of analysis output. When NDA has generated *n* pages, it terminates the analysis and prints a message on the user terminal indicating that it has done so.	/LIMIT:50
/MEMSIZ:*n*	Save file	Creates a save file 4x*n* blocks long and transfers *n*K words to it from the input file. The valid range of decimal values for *n* is from 16 to 4096. This switch is necessary only when saving dumps from secondary storage devices that have no EOFs (disk or DECtape). When the crash input resides on magtape, the save file is zero filled if the EOF is read before *n*K words have been transferred.	/MEMSIZ:124

Table 10–3 (cont.): NDA Function Control Switches

Switch	File	Description	Default
/-SP	Analysis file	Causes NDA to not spool the analysis output listing file to the line printer.	/SP
/STB	Symbols file	Identifies the file that contains both the executive and the network symbol tables for the crashed system. NDA opens the specified file and extracts the necessary symbol values. If it fails to find any required symbol values, NDA aborts the analysis and prints an error message on the terminal. If no symbol is specified via the /STB switch, CEX.STB under the current UIC is used.	CEX.STB in the current UIC
/SYM	Input file	If any required symbols are not found in the symbol table file, set their values to zero and proceed with the analysis. Normally, this condition would cause NDA to abort the analysis.	/-SYM

Chapter 11
Network Display Program (NTD)

The Network Display Program (NTD) can provide two types of displays:

- A **resource display** of the current status of the host node or any specified remote RSX/IAS DECnet node in the network. There are two formats for this display:

  Default resource display for current DECnet-RSX systems (see Figure 11-1)

  Old resource display for connections to DECnet-RSX systems having previous versions of the software (see Figure 11-2)

- A **node summary display** of all reachable nodes or a user-specified range of nodes in the network (see Figure 11-3)

You can review a continuously updated screen display on any VT100 or VT52 terminal, or you can use NTD with a hard-copy terminal to obtain a listing of the display information. The displayed information is collected by NTDEMO, a task that runs on the system for which you are displaying information.

This chapter describes the displays and outlines the commands required to invoke and customize the displays.

11.1 Invoking NTD

Use the following command to invoke NTD:

NTD [*node-name*][/PAGE=*s*]

where

node-name is a name consisting of 1 to 6 numeric or alphabetic characters, including at least 1 alphabetic character. If you do not specify *node-name*, NTD displays the status of the local node.

/PAGE=*s* is a switch that specifies the display type, where *s* is one of the following:

R Default resource display
O Old resource display
N Node summary display

This switch is most useful when running NTD from a hard-copy terminal. Without this switch, you will always get a resource display. NTD automatically checks the version of the specified node to determine whether to display the default resource display or the old resource display. From a VT100 or a VT52, you can switch to a node summary display simply by entering ESC.

Error messages that can be generated when you invoke NTD are listed in Chapter 14.

11.2 NTD Commands

NTD commands control various aspects of the real-time NTD displays. There are two modes of operation for NTD commands:

- **Immediate mode.** With the NTD display on your screen, you enter commands to directly manipulate the display — for example, to control scrolling of display information or to change to a different display.
- **Command mode.** When you switch from immediate mode to command mode, NTD displays the commands that you can enter to customize the display — for example, to change the frequency of display updates or the range of nodes for which NTD is to display information. In this mode, NTD prompts for commands. You must return to immediate mode to view the displays.

The use of each mode and its commands is defined in the following sections.

11.2.1 Immediate Mode Commands

When you invoke NTD, you automatically enter immediate mode. Enter any of the commands described in Table 11-1 to directly manipulate the screen display. The terminal's bell will ring if you enter an unrecognized command.

Table 11-1: Immediate Mode Commands

Character Key	Function
R	Changes to default resource display
O	Changes to old resource display
N	Changes to node summary display
<space bar>	Refreshes the currently selected display
D	Selects scrolling of node/circuit information only (default)
T	Selects scrolling of task information only
B	Selects simultaneous scrolling of node and task information
+	Scrolls currently selected information forward 5 items at a time for resource displays or 10 nodes at a time for node summary displays *
−	Scrolls currently selected information backward 5 items at a time for resource displays or 10 nodes at a time for node summary displays *
S	Sorts selected information in ascending order by node address instead of alphabetically by node name
H	Displays help information for immediate mode commands for 5 seconds, then redisplays the currently selected NTD status information
ESC	Enters command mode
CTRL/Z or CTRL/C	Exits NTD

* Scrolling causes information to be displayed for a range of nodes, beginning with the next available node that is five reachable addresses ahead of (or behind) the currently specified starting node. On the node summary display, a maximum of 84 nodes can be displayed on the screen at once. Unless you enter the S command (see above), nodes are displayed in alphabetic order by name. Therefore, scrolling by node address will cause the order to change.

11.2.2 Command Mode Commands

When you enter ESC to switch from immediate mode to command mode, the possible commands you can enter are displayed on your screen, followed by a command prompt. Valid commands are described in Table 11-2. Enter one command for each prompt.

When entering these commands, you need only use a single character to designate the command. For example, the commands ADDRESS=4 and A=4 both set 4 as the lowest node address for which information is to be displayed.

Table 11-2: Command Mode Commands

Command	Function
NTIME=*n*	Sets the time (in seconds) between display updates for remote nodes (default: 10 seconds)
TTIME=*n*	Sets the time (in seconds) between display updates for network tasks and circuits (default: 5 seconds)
ADDRESS=*n*	Sets the lowest node address for which information is to be displayed (range: 1 to 1023; default: node 1) *
MAXIMUM=*n*	Sets the highest node address for which information is to be displayed (range: 1 to 1023; default: node 1023) *
HOPS=*n*	Sets the maximum number of hops allowed to a remote node to be displayed (default: 31) *
COST=*n*	Sets the maximum cost allowed for a remote node to be displayed (default: 1023) *
RET	Reverts to the currently selected display and immediate mode
CTRL/Z	Exits NTD

* If you specify 0 for *n*, the value will revert to the specified default.

11.3 NTD Resource Display Format

Figures 11-1 and 11-2 are examples of the default resource display and the old resource display, respectively. The display information fields are described below.

Top of the screen

- The current date and time at the node where NTDEMO runs. This is updated approximately every second.
- The name and identification of the node being displayed. For VT100 terminals, NTD uses the VT100 graphics capability to highlight the name of the node.
- Bar graphs for the following resources:

  Logical links (LNKS)

  Communication control buffers (CCBs)

NOTE

Because CCBs can be allocated dynamically, the total number of CCBs available can vary.

Small data buffers (SDBs)

Large data buffers (LDBs)

The first set of numbers for each resource indicates the number of that resource currently in use and the total number generated into the system. For example, 12/50 in the LDB line in Figure 11-1 indicates that 12 out of 50 generated large data buffers are currently being used.

The actual graphs represented by Xs in the examples illustrate the percentage of each resource currently in use. The vertical bars on the graph (represented by ! in the figures) illustrate the highest percentage of each resource used during the current invocation of NTD. For example, in Figure 11-1, roughly 25 percent of the LDBs are currently in use, but the position of the ! shows that approximately 50 percent of the generated buffers have been in use at once during the current invocation of NTD. Bar graphs for each resource are scaled independently; for example, an X on one graph may represent usage of one of that resource, while an X on another graph may represent usage of five of that resource. Actual graphic representations can differ depending on terminal type.

At the end of the line, NTD displays the number of allocation failures for each resource type since network startup. On the LDB line, the second allocation failure field refers to receive data buffers.

The bar graph display is updated approximately every second.

```
12-JUL-83 11:41:33   Node: ELROND(19) DISTRIBUTED SYSTEMS
                       RSX-11M-Plus, Routing Node, DECnet V2.0
  LNKS    7 / 20 XXXXXXXXXXXXXX--!-----------------------   Alloc Fails:
  CCB    14 / 35 XXXXXXXXXXXXXXXX--!---------------------      0
  SDB     1 / 18 XX!---------------------------------------      0
  LDB    12 / 50 XXXXXXXXX-----------!-------------------      0 :    0

         Known Circuits:                             Network Tasks:
Circuit   Cost Size  Adj Node/Status          Task   TI   Links MBX XMT RCV

DMC-0       3  576   UNGOL (20)               NTD... CO0:    1   0   0   0
DMC-1       3        On-starting              NTDT0  TT0:    1   0   0   1
PCL-0.0     2        Off                      RVTT12 TT12:   1   0   0   1
PCL-0.1     2  576   UNGOL (20)               RVTT26 TT26:   1   0   0   1
PCL-0.2     2  576   HALDIR (18)              SEN... CO0:    1   0   1   0
PCL-0.3     2  576   SHELOB (8)               RMHACP CO0:    2   0   0   2
UNA-0       1        On
```

Figure 11-1: Sample Default Resource Display

Lower left of the screen. This data differs with the two resource display types, as described below. For either type, this information is updated at the interval specified by the NTIME command (default: every 10 seconds).

- Default resource display (see Figure 11-1). NTD displays the following information for all known circuits:

  The circuit name.

  The cost to reach the adjacent node.

  The maximum buffer size of network messages to the adjacent node (not applicable for Ethernet circuits, since each node on the Ethernet may have a different size).

  For point-to-point circuits, the name and address of the adjacent node on the circuit, if it is reachable; otherwise, the status of the circuit.

- Old resource display (see Figure 11-2). NTD displays the following information for any node that is currently reachable:

  The remote node name and its node address.

  The name of the circuit over which packets are sent to the node.

  The minimum number of hops required to reach the node and the minimum cost to reach the node. These numbers may correspond to different routes, as the least costly route may not necessarily have the fewest hops.

  The number of active logical links to that node and the current round trip delay in seconds. If the number of logical links is shown as 0, it indicates that a link has been active to that node since the network was loaded.

```
12-JUL-83 11:39:59   Node: HALDIR, HALDIR - DISTRIBUTED SYSTEMS
                     RSX-11M-Plus, Routing Node
  LNKS    4 / 20 XXXXXXXX--------!-----------------   Alloc Fails:
  CCB     1 / 38 X---------!-----------------------      0
  SDB     0 / 32 ----------------------------------      0
  LDB     1 / 39 X---------!-----------------------      0 :    0


              Remote nodes:                             Network Tasks:
     Node             Circuit   Hop/Cst LnK/Dly    Task    TI   Links MBX XMT RCV

BERGIL(6)          PCL-0.0    2/2         3/4    EVP...  C00:     0    0   0   0
SHELOB(8)          PCL-0.0    1/1                MAL.1   C00:     2    0   0   1
ORI    (9)         PCL-0.0    2/10               RMHACP  C00:     1    0   0   1
SMAUG (10)         PCL-0.0    2/2         0/4    NTD...  C00:     1    0   0   0
ORC    (11)        PCL-0.0    2/2
EOWYN (16)         PCL-0.0    2/2
ELROND(19)         PCL-0.2    1/1         1/4
ZIPPY (21)         PCL-0.0    2/5
SNARK (22)         PCL-0.0    3/6
CALDOR(23)         PCL-0.0    2/5
EXODUS(24)         PCL-0.0    2/2         0/4
ERIE   (40)        PCL-0.0    4/15
NUHAVN(46)         PCL-0.0    4/15
ADS    (50)        PCL-0.0    5/19        0/12
```

Figure 11-2: Sample Old Resource Display

Lower right of the screen. NTD displays the following information for currently active network tasks (that is, those tasks that have issued an OPN$ to the network):

- The task name.
- The TI: device associated with the task. Tasks requested through the network normally run with a TI: of CO0: for RSX-11M and M-PLUS.
- The number of logical links that are active on the task, including any outgoing connect request that has not yet completed.
- The number of entries on the task's mailbox. Entries can include:

  Incoming connect requests

  Interrupt messages

  Disconnect notifications

  Link abort notifications

  Network abort notifications

  Network events

  Refer to the *DECnet-RSX Programmer's Reference Manual* for more information about mailboxes.
- The number of outstanding logical link transmit requests.
- The number of outstanding logical link receive requests.

This information is updated at the interval specified by the TTIME command (default: every 5 seconds).

11.4 NTD Node Summary Display Format

Figure 11-3 is an example of a node summary display. The display information fields are described below.

Top of the screen

- The current date and time at the node where NTDEMO runs. This is updated approximately every second.
- The name and identification of the node being displayed. For VT100 terminals, NTD uses the VT100 graphics capability to highlight the name of the node.

Node entries

- Node name or address. By default, named nodes are listed alphabetically, followed by nodes that are identified by address only (in ascending order). You can use the immediate mode S command to order all nodes by ascending node address (see Table 11-1).
- The minimum number of hops required to reach the node and the minimum cost to reach the node. These numbers may correspond to different routes, as the least costly route may not necessarily have the fewest hops.
- The number of active logical links to the node. If the number of logical links is shown as 0, this indicates that a link has been active to that node since the network was loaded.
- The total number of nodes actually displayed. Since the screen cannot display more than 84 nodes at once, this number may not necessarily reflect the total number of nodes in a large network. You will have to use scrolling to display additional nodes. This number also does not include any nodes that have been excluded because of filtering commands (see ADDRESS, MAXIMUM, HOPS, and COST in Table 11-2).

```
12-JUL-83 11:42:01    Node: ELROND (19) DISTRIBUTED SYSTEMS
                  RSX-11M-Plus, Routing Node, DECnet V2.0
Node    Hop/Cst  Lnk  Node    Hop/Cst  Lnk  Node    Hop/Cst  Lnk  Node    Hop/Cst  Lnk
ADS      6/19         DRONE    1/1      0   MEPS     6/21         UNGOL    1/2
ALCOCK   5/13         DRTWO    1/1          MOPAC    5/15         VICKI    5/17
APPGP3   8/22         DUANE    1/1      0   NONAME   5/15         VMSRVR   1/1
APPGP4   8/22         EDAVIL   7/20         NUHAVN   5/15         VOLKS    4/11     0
BENSON   6/19         EOWYN    3/2      0   ORC      3/2          VOYAGR   6/19
BERGIL   3/2      0   ERIE     5/15         ORI      3/10     0   WAGON    5/15
BIGWIG   6/19         EWES     6/19         PENNSY   5/15         WAYBAK   7/20
BLISS    5/15         EXODUS   3/2      0   PIPPIN   1/1          WHOAMI   5/15
BNA      5/15         FILI     1/1          PLOTS    6/19         YANG     5/15
BNM      5/15         FLIWSA   7/23         PROTO8   5/13         YIN      6/19
CAD3     8/21         FRANK    1/1          RENOVO   3/2      0   YODA     3/2      0
CAD3FE   7/20         FSLENG   6/26         RUTLND   5/15         ZIGITH   5/15
CADET    5/15         HALDIR   1/2      1   SBONE    1/1      0   ZIPPY    3/5
CADLAC   5/15         HAZEL    5/15         SHELOB   2/1      0   ZORK     7/24
CADVAX   4/13         HYDRA    3/2      0   SMAUG    3/2      2   [   73]  6/19
CALDOR   3/5          KHOTEK   4/6          SNARK    4/6          [  171]  5/15
CAR      4/13         KISS     6/19         SPHINX   5/15         [  180]  4/6
CNATST   1/1          LEHIGH   5/15         TAROT    4/13         [  183]  4/3
CNJ      5/15         LIONEL   5/15         TEGATH   6/19         -Total nodes: 81-
COMET1   6/21         LYRA     5/7          TUNA     4/6
DESPRO   5/15         MANIAC   5/15         TYCO     6/19
```

Figure 11-3: Sample Node Summary Display

Chapter 12
Queue Manager (QUE)

The Queue Manager (QUE) utility is the interface to the RSX-11M V4.0 and RSX-11M-PLUS V2.0 queue manager. Any privileged user can employ QUE to initialize the File Transfer Spooler (FTS) queue and processor and to manipulate user-queued FTS requests. FTS is a DECnet-RSX utility that allows you to transfer files in much the same way as the Network File Transfer (NFT) utility, with the additional queue feature.

This chapter outlines components required by FTS and QUE and describes how to invoke and use QUE. For more information on both utilities, including error messages, see the following manuals:

- *DECnet-RSX Guide to User Utilities.* Contains a complete description of FTS
- *RSX-11M/M-PLUS Batch and Queue Operations Manual.* Contains a description of QUE user commands and error messages
- *RSX-11M/M-PLUS System Management Guide.* Contains a description of QUE system manager commands

12.1 FTS and QUE Components

Table 12-1 lists the components required by FTS and QUE and describes their relationship to queue management. Figure 12-1 diagrams the relationships among these components.

Table 12-1: FTS and QUE Components

Component	Description
FTS	The user interface that spools requests and places them on a queue.
QUE	The system manager interface that processes queue management commands.
QMG	The RSX-11 queue manager task.
FTSDEQ	The processor task that services the queue.
FAL	The File Access Listener that handles file access and transfer requests over a line at the remote node.
FTSQUE	The queue of FTS jobs that the queue manager stores in a disk area.
URBs	User request block files that contain details of the FTS requests corresponding to FTS jobs on the queue. This information contains names that FTSDEQ needs to process the requests. FTS prints this information in response to the /LIST command.
FTSSYS.LOG	The log file that contains a record of both completed and uncompleted FTS jobs.
FAL.LOG	The log file that contains a record of completed FAL jobs.

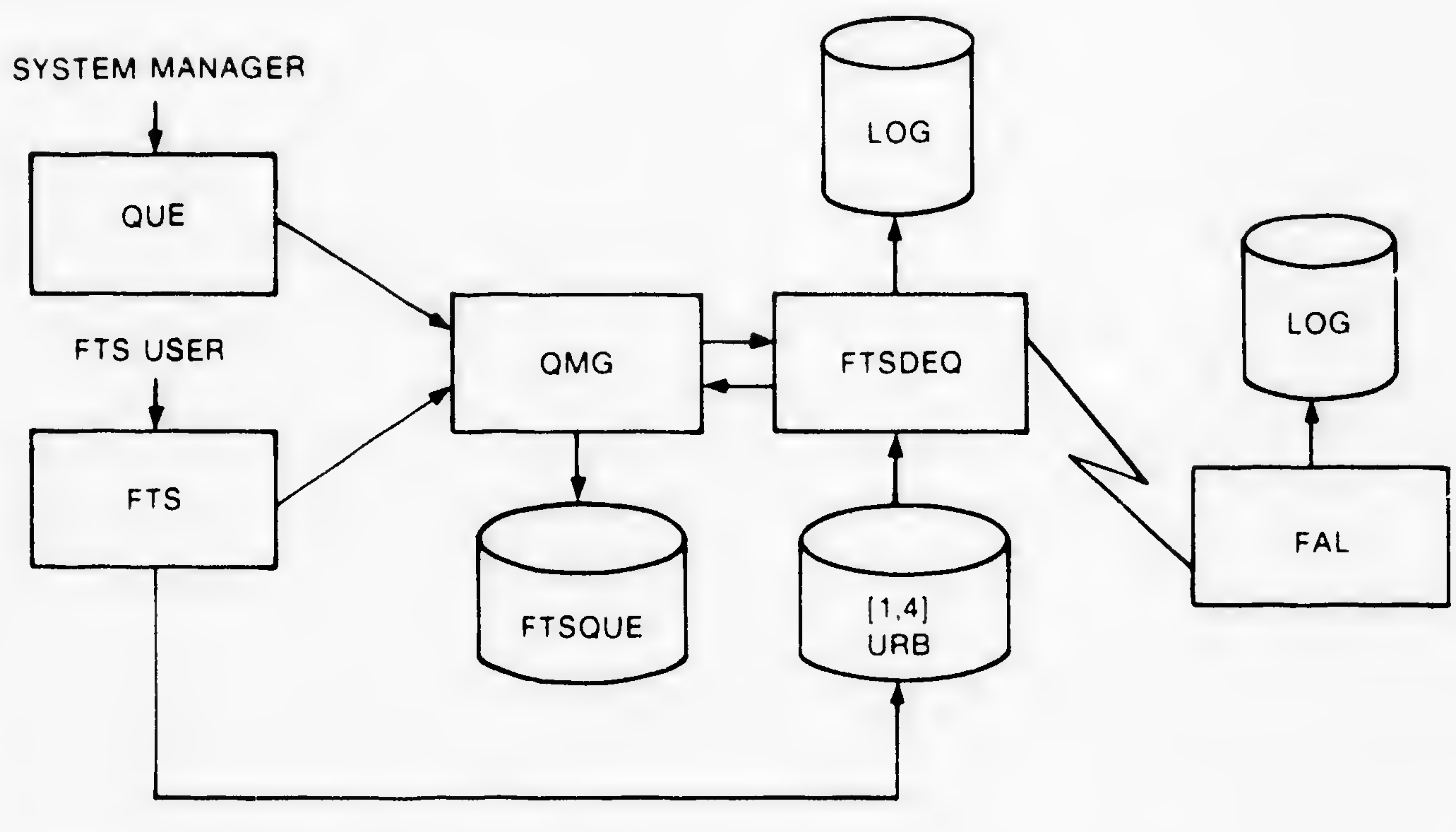


Figure 12-1: FTS and QUE Components

12.2 Invoking QUE

Use the following command to invoke QUE:

MCR>QUE *command*

where *command* is any valid QUE command. After the QUE command executes, you are returned to MCR.

12.3 QUE Command Syntax

QUE supports three command formats. Each format allows you to perform various queue management functions. These functions include:

- Initializing/deleting and starting/stopping the queue and the processor
- Assigning/deassigning the processor to/from the queue
- Holding, releasing, or deleting FTS user jobs on the queue

The following sections describe the three command formats and their functions and give examples of their use.

12.3.1 Initializing/Deleting and Starting/Stopping the Queue and the Processor

Use the following QUE command syntax and the appropriate switch to cause the queue or the processor to be initialized, deleted, started, or stopped:

MCR>QUE *name/switch*

where

name is either the queue name (FTSQUE) or the processor's installed name (FTSDEQ).

switch is one of the following switch options:

CR	creates, names, and starts the FTS queue
SP/EX	creates, names, and starts the FTS processor
STA	starts the FTS queue or processor
STO:QUE	stops the FTS queue
STO	stops the FTS processor
DEL:Q	deletes the FTS queue
UNSP	deletes the FTS processor

In an emergency situation, delete either the queue or the processor. Users can delete their jobs from the queue using FTS.

NOTE

If FTS was selected during NETGEN, the FTS queue and processor will already be set up when the network is loaded. If they were not set up by commands in file NETINS.CMD, you must create the queue and start the processor to make the FTS queue available for users.

Example:

```
MCR>QUE FTSQUE/CR
MCR>QUE FTSDEQ/SP/EX
```

The first command creates the FTS queue. The second command starts the FTS processor task.

12.3.2 Assigning and Deassigning the Processor

Use the following QUE command syntax to assign and deassign the processor to and from the queue:

MCR>QUE FTSDEQ:/*switch*:FTSQUE

where

FTSDEQ is the processor name.

FTSQUE is the queue name.

switch is one of the following switch options:

- AS assigns the processor to the queue. You must assign the processor upon initializing it in order to enable FTS operations.
- DEA deassigns the processor from the queue. Deassigning the processor prevents it from performing any functions (for example, transferring files, despooling). However, jobs can be added to the queue when the processor is deassigned. You may need to deassign the processor in special situations (for example, to prevent jobs from being completed). To reactivate the processor, use the assign switch (/AS).

There are cases where a queue must be inspected, but you do not want any of the jobs processed before listing the queue's contents. In this case, deassign the processor before listing the queue. Refer to Section 12.4 for information on related queue management functions.

12.3.3 Holding, Releasing, and Deleting FTS User Jobs

Use the following QUE command syntax to hold, release, or delete FTS user jobs on the queue:

MCR>QUE /ENT:*entry-number/switch*

where

entry-number	is the number assigned to the queue entry by the queue manager. The entry number is obtained by listing the contents of FTSQUE (see Section 12.3.2).
switch	is one of the following switch options: HO holds the specified job REL releases the specified job from the hold state DEL deletes the specified job

Ordinarily, you do not delete or hold jobs. FTS users can delete their own jobs. You need to use these commands only in special situations (for example, when you want to delay a job that is about to begin).

Example:

```
MCR>QUE /ENT:857/DEL
```

This command deletes the job with entry number 857 from the queue.

12.4 Related Queue Management Functions

In addition to using QUE for queue management, you should periodically perform other related queue management functions. These functions include:

- Purging the FTS system log file (FTSSYS.LOG) and the FAL system log file (FAL.LOG)
- Listing the contents of FTSQUE and of FTS jobs
- Listing and purging the user request block (URB) files

The following sections describe procedures for performing these functions.

NOTE

FTS will retry requested file operations if they fail due to network problems. After a number of unsuccessful retries (default = 5), the FTS job will be put into the hold state in FTSQUE. The user must decide whether to release the job for one more retry or to delete (abort) it.

12.4.1 Purging FTSSYS.LOG and FAL.LOG

Periodically, you should purge the FTS system log file (FTSSYS.LOG) and the FAL system log file (FAL.LOG). FTSSYS.LOG contains a record of every job FTS has received, along with coded status information, time of receipt, and time of completion. The file is useful for statistical purposes. FAL.LOG records network file transfer information for each file copied over the network.

FTSDEQ automatically deletes URB file entries as jobs complete. However, it does not delete FTS system log file entries. Use the Peripheral Interchange Program (PIP) to purge these files when necessary. (See the *RSX-11M/M-PLUS Utilities Manual* for more information.) The following PIP command displays the FTS system log file:

```
>PIP TI:=[1,4]FTSSYS.LOG
                         .
                         .
                         .
```

Refer to the *DECnet-RSX Guide to User Utilities* for a description of the system log file display.

During network generation, you can choose to create the system log file in UFD [1,4] with no file protection. Optionally, log entries are copied to a log file in the UFD of the user who queued the request. The user is expected to maintain this file.

12.4.2 Listing FTSQUE and FTS Jobs

Periodically, or when necessary, list the contents of FTSQUE and of FTS jobs. The following command lists the contents of FTSQUE:

```
MCR>QUE FTSQUE:/LI:P
                    .
                    .
                    .
```

The system displays:

```
 ** PRINT QUEUES **
 FTSQUE =>FTSDEQ
[1,1]     223771 ENTRY:857        PRINT AFTER 2-FEB-83 11:37
          1 DB0:[1,4]FTS4166.URB;1
```

The FTSQUE contains one job, number 223771. Note that FTSDEQ is the processor for FTSQUE. This field appears only when the processor has been assigned to the queue. If this field does not appear, the processor is deassigned, but there can still be entries in the queue. The job was originally queued from the UIC [4,166]. After it failed, the job was requeued by the FTSDEQ task from UIC [1,1]. The queue manager entry number is 857. The job is to be printed (in this case retried) at 11:37.

Note the format of the URB file specification:

device:[*ufd*]FTS*nnnnnn*.*URB*;xx

where

nnnnnn is the user's current UIC (with leading zeroes suppressed).

xx is the request number. If multiple jobs are queued from a single UIC, this number is greater than 1.

If, after inspecting the FTS queue, you want more information on a user's job on the queue, type:

```
>SET /UIC=[user uic]
>FTS /LI
```

The system displays:

```
FTS USER REQUEST LISTING                              2-FEB-83 11:45

223553  TLR/50                         /IM/LO
 Queued 2-FEB-83 11:45                                Retried 0 times
 Source File -- DB0:[4,122]NODES.SYS;4
 Destination File -- XYZ::[4,131]NODES.SYS

223573  TLR/50                         /IM/LO
 Queued 2-FEB-83 11:45                                Retried 0 times
 Source File -- DB0:[4,122]NODFIL.MAC;10
 Destination File -- XYZ::[4,131]NODFIL.MAC

223507  TLR/50                         /IM/LO
 Queued 2-FEB-83 11:45                                Retried 0 times
 Source File -- DB0:[4,122]PRMFIL.MAC;1
 Destination File -- XYZ::[4,131]PRMFIL.MAC
```

The job numbers at the beginning of each entry are the only cross-reference you have to the system queue list. For more information on the FTS /LI switch, see the *DECnet-RSX Guide to User Utilities*.

12.4.3 Listing and Purging URB Files

Periodically, or when necessary, list and purge the URB files. The following PIP command produces a display of the URB directory:

```
>PIP [1,4]*.URB;*/LI
                    .
                    .
                    .
```

If you have deleted a job with QUE, you must also delete its associated URB file or files. QUE does not do this automatically. However, when an FTS user deletes a job, it also is deleted automatically from its URB file.

Chapter 13
Trace Interpreter Task (TRI)

The trace interpreter task (TRI) is part of the trace facility provided for RSX-11 PSI users. This facility enables you to diagnose software problems on a PSI line connected to a device. The facility traces message frames passing between the X.25 level 2 protocol and the device driver and copies the traced frames to a disk file. The contents of this file can then be analyzed and printed.

Use NCP TRACE commands to control the tracing of frames and to create the file (see the CLEAR TRACE, SET TRACE, and SHOW TRACE commands in Chapter 6). Then use TRI to analyze the contents of the trace disk file.

13.1 Invoking TRI

There are three methods of invoking TRI:

1. `MCR>TRI`

   TRI then prompts

   `TRI>`

   Enter your TRI command immediately following the prompt. If the command executes, the utility prompt is displayed on the next line. If the command is unsuccessful, an error message is displayed on the next line indicating the reason for the error followed by the utility prompt.

2. `MCR>TRI` *command*

   where *command* is any valid TRI command. With this method, you return to MCR after the TRI command is executed.

3. `MCR>TRI @`*command-file*

   where *command-file* is the name of an indirect command file containing TRI commands. You can nest command files.

13.2 Exiting TRI

To exit TRI, enter CTRL/Z or RET.

13.3 TRI Command Format

The TRI command line has the following format:

[*outfile*[/*switches*]=][*infile*[/*switches*]]

where

outfile is the listing file. If not specified, this defaults to the input file name with file type .LST.

infile is the name of the file containing the tracing data. If this name is not specified, this file defaults to LB:[1,6]PSITRACE.SYS.

switches are the options available for controlling the form and content of the listing produced by TRI. Any compatible combination of the switches shown in Table 13-1 can be specified on either the input file or the output file.

Table 13-1: Trace Interpreter Switches

Switch	Meaning	Comments
/SP	Spool the output file	Default switch. (Output is spooled unless /-SP is specified.)
/-SP	Do not spool the output file	
/OC *	Output user data in octal	Cannot be specified with /-IN.
/AS *	Output user data in ASCII	Cannot be specified with /-IN.
/IN	Interpret trace data	Default switch. (Data is interpreted unless /-IN specified.)
/-IN	Do not interpret trace data	Only /SP or /-SP can be specified with this switch.
/LI:*n*	Output trace data for system line *n* only	The *n* is octal. If you want to specify a decimal number, use *n*. instead. The default for *n* is 0. Cannot be specified with /-IN.
/CH:*n*	Output trace data for channel number *n* only	The *n* is octal. If you want to specify a decimal number use *n* instead. The default for *n* is 0. Cannot be specified with /-IN.

* If you specify both /OC and /AS, trace data is printed in octal bytes with the corresponding ASCII characters underneath.

Examples:

```
MCR>TRI =/OC/AS
```

This command uses the /OC and /AS switches to output user data in octal and ASCII characters.

```
MCR>TRI =/LI:1/OC
```

This command uses the /OC switch to output octal user data for system line 1 only.

13.4 Contents of Trace Output

TRI reads the trace file created by the tracing facility and lists the following information (depending upon the frame type) for each traced frame:

- Any error status from the CCB
- The system line number
- The direction of transfer (Rx or Tx)
- The frame address byte
- Setting of the P/F bit
- N(S) and/or N(R) numbers
- The frame type as text (for example, SABM)
- The logical channel identifier
- Settings of the Q, D, and M bits
- The packet type as text (for example, Call Req.)
- P(S) and/or P(R) numbers
- Cause and diagnostic fields
- The first few bytes of user data

13.5 TRI Error Messages

TRI prints an error message on your terminal when it detects an operating error. All TRI error messages are listed in Chapter 14.

PART IV
ERROR MESSAGES

This section summarizes all error messages for the following tools and utilities:

CCR	NTD
CFE	NTINIT
HLD	NTL
NCP	TRI
NDA	VNP

Chapter 14
Utility Error Messages

This chapter lists the error messages sent by the following utility programs:

- Console carrier requester (CCR)
- Configuration File Editor (CFE)
- Host Task Loader (HLD)
- Network Control Program (NCP)
- Network Dump Analyzer (NDA)
- Network Display Program (NTD)
- Network initializer (NTINIT)
- Network loader (NTL)
- Trace interpreter task (TRI)
- Virtual Network Processor (VNP)

You would not be surprised to receive a CFE error message if you had just invoked a CFE command. However, CFE, NCP, and VNP use the NTINIT and NTL utilities to process your commands. Therefore, an error could occur in one of those utilities instead of in the one that you invoked. You will receive an error message from the utility in which the error occurred. Each error message has a prefix that identifies the utility that sent it.

In this chapter, error messages are arranged in alphabetical order by utility and message text. Variable fields within messages are ignored for purposes of alphabetization.

14.1 CCR Error Messages

CCR — Component in wrong state, xxx

The specified component was not in the correct state to proceed with the operation of console carrier (*xxx* = circuit or system).

CCR — File I/O error, xxx

An I/O error occurred while trying to read the specified file (*xxx* = secondary loader or load file).

CCR — File open error, xxx

CCR was unable to open the specified file (*xxx* = secondary loader or load file).

CCR — Invalid file contents, xxx

The contents of the specifie ! file are not in the required format (*xxx* = secondary loader or load file).

CCR — Invalid identification format, xxx

The format for the specified parameter is invalid (*xxx* = node, service circuit, hardware address, or service password).

CCR — Line communication error

The operation failed due to a communication error on the line.

CCR — Line protocol error

The operation failed due to a protocol error.

CCR — Management programming error

An internal programming error has been detected.

CCR — Message format error

The format of a received protocol message was invalid.

CCR — Parameter missing, xxx

The required parameter was not defined in the command line and the down-line load data base (*xxx* = node, service circuit, hardware address, or service password).

CCR — Privilege violation

The user does not have sufficient privileges to run CCR.

CCR — Protocol/address pair in use

The protocol/address pair used to communicate with the target system is already in use.

CCR — Remote console already in use

Another user on another node has already reserved the remote console on the target system.

14.2 CFE Error Messages

All CFE error messages documented in this chapter are classified according to severity and cause. Both classifications are described below and are noted at the beginning of each error description.

Severity classifications

Informational	There is a minor problem, but it does not affect the task's exit status.
Warning	Minor error. Processing continues.
Error	Major error. The command was accepted, but failed to execute completely. The system waits for a new command.
Severe Error	The command was not accepted. Processing stops and control returns to the monitor.

Cause classifications

CETAB Parse	These errors occur during the parsing of CETAB.MAC. They usually occur when someone has edited CETAB using a text editor. Restore CETAB to its original content.
CETAB	These errors are generated in response to valid commands. They usually occur when someone has edited CETAB using a text editor. Restore CETAB to its original content.
Configuration	These errors are generated by valid commands that refer to features or components that are not in the CETAB file being edited. Check the CETAB file and enter the correct command.
Command	These errors usually are caused by a mistyped command or by an incorrect or inapplicable parameter. Correct the command and reenter it.
General	These errors usually indicate a problem with the CFE installation or with the HELP or CETAB files. The cause is probably an invalid UIC or something similar.

CFE — Access verification not supported

(Error/Configuration) You do not have access verification supported in your system, so you cannot change the verification state.

CFE — Bad port value

(Severe Error/CETAB Parse) The number of ports on the PSN$DF macro is out of range or contains nonnumeric characters.

CFE — Bad transport subaddress value

(Severe Error/CETAB Parse) The transport subaddress values on the PSN$DF macro are out of range or contain invalid characters.

CFE — Buffer allocation failure

(Error/Command) CFE tried to allocate an internal buffer to complete the command but failed. If you must make a new entry in CETAB, then remove CFE and reinstall it with an increment.

CFE — Call mask and call value are different lengths

(Error/Command) The call mask and call value of a destination must be of the same length.

CFE — Circuit not in configuration

(Error/Configuration) The circuit in the command is not in the data base.

CFE — Command line error

(Error/Command) The command has a bad character or equivalent.

CFE — Consistency error

(Severe Error/General) There is an internal consistency error in CFE. Please contact your software specialist to determine the cause of this error.

CFE — Destination not in configuration

(Error/Configuration) The destination in the command is not in the data base.

CFE — DTE not in configuration

(Error/Configuration) The DTE in the command is not in the data base.

CFE — DTE number ignored for incoming SVC: DLM-*u.t*

(Warning/Command) The DTE parameter is not applicable to incoming SVCs and is ignored. The remainder of the command will be processed.

CFE — DTE number missing from outgoing SVC: DLM-*u.t*

(Severe Error/CETAB Parse) The DTE number is required for outgoing SVCs. Reenter the command with the DTE number.

CFE — DTE with DLM circuits not purged: *DTE-address*

(Error/Command) The indicated DTE was not purged because some of its circuits are data-link-mapping circuits. Data-link-mapping circuits may not be purged.

CFE — Duplicate PSN$DF in CETAB

(Severe Error/CETAB Parse) There is more than one PSN$DF macro in the CETAB.MAC file. Remove one of the PSN$DF macros.

CFE — Error in call mask or call value

(Error/Command) The format in the call mask or call value in the command is invalid.

CFE — Error in channel list

(Error/Command) The format of the channel list in the command is invalid.

CFE — Error in channel number

(Severe Error/CETAB Parse) The channel number on a CHN$DF macro is out of range or contains nonnumeric characters.

CFE — Error in controller number

(Severe Error/CETAB Parse) The controller number on an SLT$DF macro is invalid.

CFE — Error in DDM name

(Severe Error/CETAB Parse) The DDM name on an SLT$DF macro is too long.

CFE — Error in DLC name

(Severe Error/CETAB Parse) The DLC name on an SLT$DF macro is too long.

CFE — Error in DTE address

(Severe Error/CETAB Parse) The DTE address field on an SVC$DF macro is out of range.

CFE — Error in flags

(Severe Error/CETAB Parse) The flags field on a CUGDF, PSNDF, PVCDF, SLTDF, or SVC$DF macro is unrecognized or invalid.

CFE — Error in line ID

(Severe Error/CETAB Parse) The line identification on a DTE$DF macro is of an invalid format.

CFE — Error in line type

(Severe Error/CETAB Parse) The line type on an SLT$DF macro is unrecognized.

CFE — Error in LLC name

(Severe Error/CETAB Parse) The LLC name on an SLT$DF macro is too long.

CFE — Error in maximum circuits

(Severe Error/CETAB Parse) The maximum circuits value on an X2P$DF macro is out of range or has an invalid format.

CFE — Error in maximum recalls

(Severe Error/CETAB Parse) The maximum recall field on an SVC$DF macro is out of range.

CFE — Error in network name

(Severe Error/CETAB Parse) The network name on a DTE$DF or PSN$DF macro is too long or contains invalid characters.

CFE — Error in opening file

(Severe Error/General) CFE could not open the specified CETAB source file. Make sure that the file is where you specified, that the file is not locked, and that you have write privileges to the file.

CFE — Error in owner

(Severe Error/CETAB Parse) The owner process field on an SLT$DF macro is unrecognized.

CFE — Error in packet size

(Severe Error/CETAB Parse) The packet size field on an SVC$DF or X2P$DF macro is out of range.

CFE — Error in recall timer

(Severe Error/CETAB Parse) The recall timer value field on an SVC$DF macro is out of range.

CFE — Error in SVC name

(Severe Error/CETAB Parse) The SVC circuit ID on an SVC$DF macro has an invalid format.

CFE — Error in unit number

(Severe Error/CETAB Parse) The unit number on an SLT$DF is invalid.

CFE — Error in window size

(Severe Error/CETAB Parse) The window size on an SVC$DF or X2P$DF macro is out of range.

CFE — Error writing file

(Severe Error/General) CFE was unable to write a record to the output file; therefore the CETAB source file was not updated. Check to make sure that the disk is not full.

CFE — Executor node not allowed

(Error/Command) An attempt has been made to define the executor node as a remote node.

CFE — Executor node not allowed with command

(Error/Command) The command parameters specified in the command are not permitted on the executor node.

CFE — Group not in configuration

(Error/Configuration) The closed user group in the command is not in the data base.

CFE — Illegal baud rate

(Error/Command) The baud rate specified in the command is out of range or not in the table of valid baud rates. Reenter the command with a legal baud rate.

CFE — Illegal call mask

(Severe Error/CETAB Parse) The call mask on a DSC$DF macro is too long or contains invalid characters.

CFE — Illegal call value

(Severe Error/CETAB Parse) The call value on a DSC$DF macro is too long or contains invalid characters.

CFE — Illegal circuit type

(Error/Command) The circuit type in the command is illegal for the requested operation. For example, it is illegal to purge a DDCMP circuit.

CFE — Illegal command

(Error/Command) The command you gave to CFE is incorrect.

CFE — Illegal CUG name

(Severe Error/CETAB Parse) The user group name on a CUG$DF or DSA$DF macro is too long or contains invalid characters.

CFE — Illegal destination address

(Severe Error/CETAB Parse) The destination address on a DSA$DF macro is not a valid DTE address.

CFE — Illegal destination name

(Severe Error/CETAB Parse) The destination name on a DST$DF macro is too long or contains invalid characters.

CFE — Illegal destination priority

(Severe Error/CETAB Parse) The list insertion priority on a DST$DF macro is out of range or contains nonnumeric characters.

CFE — Illegal destination type

(Severe Error/CETAB Parse) The destination type on a DST$DF macro is not legal for PSI.

CFE — Illegal device priority

(Error/Command) The device priority specified in the command is out of range.

CFE — Illegal DTE name

(Severe Error/CETAB Parse) The DTE name on a RDT$DF macro is too long or contains invalid characters.

CFE — Illegal executor node address

(Error/Command) An attempt was made to define the executor node address to greater than the maximum address for the network.

CFE — Illegal group number

(Severe Error/CETAB Parse) The group number on a CUG$DF macro is out of range or contains nonnumeric characters.

CFE — Illegal logical channel number

(Severe Error/CETAB Parse) The logical channel number on a PVC$DF macro is out of range or contains nonnumeric characters.

CFE — Illegal maximum data size

(Severe Error/CETAB Parse) The maximum data size on a PVC$DF macro is out of range or contains nonnumeric characters.

CFE — Illegal object number

(Severe Error/CETAB Parse) The object number on a DST$DF macro is out of range or contains nonnumeric characters.

CFE — Illegal owner process

(Severe Error/CETAB Parse) The owner process identification field on a PVC$DF macro is not a recognized owner.

CFE — Illegal PVC name

(Severe Error/CETAB Parse) The circuit identification on a PVC$DF macro is too long or contains invalid characters.

CFE — Illegal state

(Severe Error/CETAB Parse) The state on a PVC$DF macro is not a recognized state.

CFE — Illegal subaddress high range

(Severe Error/CETAB Parse) The indicated subaddress on a DSA$DF macro is out of range or contains nonnumeric characters.

CFE — Illegal subaddress low range

(Severe Error/CETAB Parse) The indicated subaddress on a DSA$DF macro is out of range or contains nonnumeric characters.

CFE — Illegal task name

(Severe Error/CETAB Parse) The task name on a DST$DF macro is too long or contains invalid characters. This message is also generated if a task name is included and the destination object number is not zero.

CFE — Illegal window size

(Severe Error/CETAB Parse) The window size on a PVC$DF macro is out of range or contains nonnumeric characters.

CFE — Insufficient buffer space

(Severe Error/General) CFE was not built with enough buffer space to accommodate the CETAB source file information. Remove CFE and reinstall it with an increment.

CFE — Invalid block size

(Severe Error/CETAB Parse) The block size on an X2P$DF macro is out of range or of an invalid format.

CFE — Invalid call timer value

(Severe Error/CETAB Parse) The call timer field on the X3P$DF macro is out of range or is of an invalid format.

CFE — Invalid circuit identification

(Error/Command) The format of the circuit ID in the command is invalid.

CFE — Invalid clear timer value

(Severe Error/CETAB Parse) The clear timer field on the X3P$DF macro is out of range or is of an invalid format.

CFE — Invalid counter timer value

(Severe Error/CETAB Parse) The counter timer value on a DTEDF, PVCDF, or SLT$DF macro is out of range or contains invalid characters.

CFE — Invalid default block size

(Severe Error/CETAB Parse) The default block size on the X3P$DF macro is out of range or is of an invalid format.

CFE — Invalid default window size

(Severe Error/CETAB Parse) The default window size field on the X3P$DF macro is out of range or is of an invalid format.

CFE — Invalid destination sequence

(Error/CETAB) An invalid destination sequence has been detected. This usually indicates a corrupt CETAB.MAC file.

CFE — Invalid DTE address

(Severe Error/CETAB Parse) The DTE address on a DTE$DF or RDT$DF macro is too long or contains invalid characters.

CFE — Invalid DTE state

(Severe Error/CETAB Parse) The state on a DTE$DF macro is not a recognized DTE state.

CFE — Invalid hash table size

(Severe Error/CETAB Parse) The hash table size on a DTE$DF macro is out of range, is not a power of 2, or contains invalid characters.

CFE — Invalid holdback timer value

(Severe Error/CETAB Parse) The holdback timer value on an X2P$DF macro is out of range or is of an invalid format.

CFE — Invalid line identification

Two error conditions can display this message:

- (Severe Error/CETAB Parse) CFE found an X2P$DF macro containing an error in the line identification. This error occurs during the initial scan of the CETAB file when CFE is invoked.
- (Error/Command) The line ID specified in the command is incorrect.

CFE — Invalid maximum block size

(Severe Error/CETAB Parse) The default block size on the X3P$DF macro is out of range or is of an invalid format.

CFE — Invalid maximum clears value

(Severe Error/CETAB Parse) The maximum clears field on the X3P$DF macro is out of range or is of an invalid format.

CFE — Invalid maximum resets value

(Severe Error/CETAB Parse) The maximum resets field on the X3P$DF macro is out of range or is of an invalid format.

CFE — Invalid maximum restarts value

(Severe Error/CETAB Parse) The maximum restarts field on the X3P$DF macro is out of range or is of an invalid format.

CFE — Invalid maximum window size

(Severe Error/CETAB Parse) The maximum window size field on the X3P$DF macro is out of range or is of an invalid format.

CFE — Invalid multicopy task name

(Error/Command) The task name of a multicopy object must be of the format *xxx*$$$, where *xxx* is 1 to 3 alphanumeric characters. This message is generated when an attempt is made to define a multicopy object without a multicopy name or a single copy object with a multicopy name.

CFE — Invalid number

(Error/Command) The command has a number out of range.

CFE — Invalid process identification

(Error/Command) The process name specified in the command is too long.

CFE — Invalid reset timer value

(Severe Error/CETAB Parse) The reset timer field on the X3P$DF macro is out of range or is of an invalid format.

CFE — Invalid restart timer value

(Severe Error/CETAB Parse) The restart timer field on the X3P$DF macro is out of range or is of an invalid format.

CFE — Invalid retransmit count

(Severe Error/CETAB Parse) The retransmit count on an X2P$DF macro is out of range or is of an invalid format.

CFE — Invalid retransmit timer value

(Severe Error/CETAB Parse) The retransmit timer value on an X2P$DF macro is out of range or is of an invalid format.

CFE — Invalid wildcard syntax

(Error/Command) The wildcard line or circuit ID in the command is of an invalid format.

CFE — Line not in configuration

(Error/Configuration) The line specified in the command is not in the CETAB source file.

CFE — Logging monitor not supported

(Error/Configuration) The logging monitor is not supported in your system configuration.

CFE — Missing LLC$DF for NW

(Error/CETAB) The LLC$DF macro for NW is missing from the data base. This usually indicates a corrupt CETAB.MAC file.

CFE — Missing parameter: *parameter*

(Error/Command) The indicated parameter is required for the type of operation requested and is missing from the command. Correct the command and reenter it.

CFE — Missing X2P$DF for line *line-id*

(Error/CETAB) There is no X2P$DF macro for the indicated line. This usually indicates a corrupt CETAB.MAC file.

CFE — No buffer definition

(Error/CETAB) The CETAB source file does not contain a BUF$DF macro defining your buffers.

CFE — No HELP available

(Informational/General) CFE could not find the help file.

CFE — No node definition

(Error/CETAB) The CETAB source file does not contain a NOD$DF macro defining the local node.

CFE — No partition definition

(Error/CETAB) The CETAB source file does not contain a PAR$DF macro defining the network pool.

CFE — No PSI support in system

(Error/Configuration) The CETAB being processed does not include support for PSI.

CFE — No routing definition

(Error/CETAB) The CETAB source file does not contain a ROU$DF macro defining the routing parameters.

CFE — No station template for *line-id*

(Error/CETAB) There are no station templates for the indicated line. This usually indicates a corrupt CETAB.MAC file.

CFE — No unit CSR on this line

(Error/Command) An attempt was made to change the unit CSR for a line that does not have a unit CSR. This part of the command was ignored.

CFE — No XPT features macro

(Error/CETAB) The CETAB source file does not contain an FEA$DF macro defining the XPT features. You will not be able to modify any of the states defined by this macro.

CFE — No XPT process macro

(Error/CETAB) The CETAB source file does not contain an LLC$DF macro defining the XPT process.

CFE — No X.29 support in configuration

(Error/Configuration) The CETAB.MAC file being processed does not contain the X29$DF macro.

CFE — No X3P$DF in data base

(Error/CETAB) The X3P$DF macro is not in the data base. This usually indicates a corrupt CETAB.MAC file.

CFE — Node address already in use

(Error/Command) The address you want to give this node is already in use.

CFE — Node name already in use

(Error/Command) The name you want to give this node is already in use.

CFE — Node not in configuration

(Error/Configuration) The node specified in the command is not in the CETAB source file.

CFE — Number of SDBs exceeds number of CCBs

(Warning/CETAB) The maximum number of SDBs specified in the BUF$DF macro exceeds the number of CCBs specified in the BUF$DF macro.

CFE — Object not in configuration

(Error/Configuration) The object specified in the command is not in the CETAB source file.

CFE — Object task name must be specified

(Error/Command) An attempt was made to create an object type entry without an associated task name. Reenter the command with the task name supplied.

CFE — Operation invalid for line type

(Error/Command) The requested operation is not applicable to the line type indicated in the command.

CFE — Parameter applicable to PSI systems only

(Error/Configuration) One or more of the parameters in the command is inapplicable to systems without PSI, and the CETAB being processed is for a non-PSI system.

CFE — Parameter illegal for circuit type

(Error/Command) One or more of the parameters in the command cannot be set for the type of circuit indicated.

CFE — Parameter illegal for wildcard operation

(Error/Command) One or more of the parameters in the command is not allowed for wildcard or KNOWN operations.

CFE — Parameter not applicable

(Error/Configuration) One or more of the parameters in the command is not applicable to the CETAB configuration (for example, routing parameters for an end node).

**CFE — Pool byte extension less than minimum
Defaulting to *nnn*.**

(Warning/Command) The pool byte extension on the command is less than the minimum required for structures defined in the CETAB. The minimum value *nnn*. is being used instead.

CFE — Process not in configuration

(Error/Configuration) The process name specified in the command is not in the CETAB source file.

CFE — Routing parameters not applicable

(Error/Command) The routing parameters in the command are not applicable to an end node.

CFE — Syntax error

(Severe Error/CETAB Parse) The indicated macro contains an invalid or unrecognized value.

CFE — Task name not allowed with object type zero

(Error/Command) An attempt was made to force a single task to be associated with the general task connect (type zero). This part of the command was ignored.

CFE — Too many stations

(Severe Error/CETAB) There are more stations on a particular line than CFE is prepared to handle. This message is followed by the syntax error message and the entry read from CETAB that caused the station number overflow.

CFE — Unknown HELP qualifier

(Informational/General) CFE has no help on the qualifier requested.

CFE — Usage cannot be permanent

(Error/Command) The usage of an SVC must be either incoming or outgoing.

CFE — XPT not allowed to load line tables in process space

(Error/Configuration) The command included an extension for line tables for XPT. Since XPT cannot have line tables in process space, this part of the command was ignored.

CFE — X25-SERVER not in configuration

(Error/Configuration) There is no support for the X.25 server module in the system.

CFE — X29-SERVER not in configuration

(Error/Configuration) There is no support for the X.29 server module in the system.

14.3 HLD Error Messages

Some of the HLD error messages include strings that are replaced with run-time variables. A list of these strings and the type of information they contain is shown below.

(*-nnn.*)	A DSW or IOSB error code (always negative).
(*nnn.*)	A network code (always positive).
node	The name of the remote RSX-11S node.
task	The installed task name received from SLD.
filespec	A copy of the file specification from the HTASK$ macro. This is followed by (File) to indicate that HLD found a match from the external mapping tables, or by (G-P) to indicate that the task was found on the general purpose list, or by (File, G-P) to indicate that both conditions were satisfied.

Variable fields are ignored in the alphabetization of error messages in this chapter.

HLD — *node/task* — Base address not a multiple of 4K
HLD — *filespec*

The file was not built with the /MM switch. Rebuild the file.

HLD — *node/task* — Connect accept failure (*-nnn.*)
HLD — *filespec*

The attempt to accept a connection from a remote SLD has failed. The remote node may have become unreachable.

HLD — *node/task* — Disk read failure (*-nnn.*)
HLD — *filespec*

This error message indicates a disk error while reading the file (initial load or checkpoint read). Validate the integrity of the file and the disk. The second error message line may or may not appear.

HLD — *node/task* — Disk write failure (*-nnn.*)

This error message indicates a disk error while writing to the file (checkpoint write). Validate the integrity of the file and the disk.

HLD — *node/task* — File has never been installed

The task file has not been installed (via VMR) into the RSX-11S system. This error message appears if you rebuild the task file but neglect to install it. Use VMR to install the task into the RSX-11S system.

HLD — *node/task* — File has no task header
HLD — *filespec*

The file was built with the /-HD switch. It is not a valid RSX-11S task image. Rebuild the file.

HLD — *node/task* — File is larger than partition
HLD — *filespec*

The initial load size of the file is larger than the partition size in the RSX-11S system. Either make the partition larger or rebuild the file to use a smaller partition size.

HLD — *node/task* — File is not contiguous
HLD — *filespec*

The file is not contiguous. Either rebuild the file or use PIP to make the file contiguous (see example below).

```
>PIP /NV/CO=filespec
>PIP filespec/PU
```

HLD — *node/task* — File spec parse error
HLD — *filespec*

The file specification for the file has an invalid format. Rebuild the HLD data base.

HLD — *node/task* — G-P tasks may not be checkpointed
HLD — *filespec*

A checkpoint operation was attempted for an HTASK$ macro from the general purpose list. Edit and rebuild the appropriate mapping table to move the HTASK$ macro from the general purpose list into a node-specific list.

HLD — Invalid connect request from *node-id*

A task attempted to connect to HLD but did not supply a valid optional connect data message. Refer to your object type tables within NETACP. HLD is connected to by the object NO.RTL and not by the installed task name. NO.RTL is defined in the NSSYM$ macro. Its object type number is 18.

HLD — *node/task* — Invalid data type in mailbox (*nnn.*)

A network data type was found in the mailbox that is not used by HLD/SLD. If the data type is a connect, NSP has tried to place more connects onto the mailbox than HLD can handle.

HLD — *node/task* — Label block read failure (-*nnn*.)
HLD — *filespec*

The label block (TKB information) could not be accessed successfully. Rebuild the file.

HLD — *node/task* — LUN fixing not supported
HLD — *filespec*

The remote SLD was not built with the LUN-fixing module. Either rebuild the remote SLD, or edit and rebuild the appropriate mapping table to move the HTASK$ macro from general purpose list into a node-specific list. This error also occurs if the HTASK$ macro is already in a node-specific list but uses the LUN control argument.

HLD — Mailbox LUN assignment failure (-*nnn*.)

HLD could not successfully assign the network mailbox LUN to the NS: device. Check to see if the local node's state is ON.

HLD — *node/task* — Mapping file base address not zero

This error is caused by an incorrect TKB command file. Edit the TKB command file, add the /-MM switch to the output file specifier, and then add or modify the PAR statement to read PAR=ANY:0:20000. Then rebuild the file.

HLD — *node/task* — Mapping file contains a task header

This error is caused by an incorrect TKB command file. Edit the TKB command file and add the /-HD switch to the output file specifier. Then rebuild the file.

HLD — *node/task* — Mapping file contains garbage

HLD can successfully read the external file, but the file is internally inconsistent. Rebuild the file.

HLD — *node/task* — Mapping file is larger than HLDFIL buffer space

The external mapping file is larger than the buffer space available. Merge the external table into the internal table. Discontinue use of the external table.

HLD — *node/task* — Mapping file is not contiguous

The file is not contiguous. Either rebuild the file or use PIP to make the file contiguous (see example below).

```
>PIP /NV/CO=HLDTAB.TSK
>PIP HLDTAB.TSK/PU
```

HLD — *node/task* — Mapping file label block read failure (-*nnn*.)

HLD could not read the file's label block (which contains TKB descriptive information). Rebuild the file.

HLD — *node/task* — Mapping file open failure (-*nnn*.)

HLD could not open the file. Rebuild the file.

HLD — *node/task* — Mapping file read failure (-*nnn*.)

HLD could not read the external mapping file. Rebuild the file.

HLD — Network data queue empty

HLD found nothing in the network data queue. This error occurs if you issue MCR>RUN HLD... or if the connection times out before HLD can remove the connect request from the network data queue. Increase the connect timeout period.

HLD — *node/task* — Network disconnect (*nnn*.)

NSP has broken the logical link before the transfer has been completed.

HLD — Network open failure (-*nnn*.)

HLD could not successfully issue the OPN$ macro. Check to see if the local node's state is ON.

HLD — *node/task* — Network read failure (-*nnn*.)

A communication error has occurred at the remote SLD. The logical link is broken.

HLD — Network SPA failure (-*nnn*.)

HLD could not successfully issue the SPA$ macro.

HLD — *node/task* — Network write failure (-*nnn*.)

A communication error has occurred at the remote SLD. The logical link is broken.

HLD — *node/task* — No entry in *node/task* tables

Your mapping tables do not contain an HTASK$ macro that corresponds to the task name from the error message.

HLD — NS: Work LUN assignment failure (-*nnn*.)

HLD could not successfully assign one of its nonmailbox LUNs to the NS: device.

Error Messages

HLD — *node/task* — Open failure (-*nnn.*)
HLD — *filespec*

The file named by the HTASK$ macro could not be opened successfully. If there is an FCS problem, you must rebuild the file. If not, change either the file or the mapping table. If you change the file name, make it match the mapping table. If you change the mapping table, make it match the file name.

HLD — *node/task* — Partition is larger than CHKPT *space*
HLD — *filespec*

The partition size in the RSX-11S system is larger than the checkpoint space inside the file. Typically, this indicates that the partition size in your PAR= statement is smaller than the partition's real size in the RSX-11S system. Although the load size of a task can be much smaller than its partition, the entire partition is transferred during checkpoint operations. Rebuild the file with the same partition size as the RSX-11S system.

HLD — *node/task* — SLD abort/disconnect

This error message indicates that SLD detected a file error during an initial load or checkpoint read. Reinstall the task into the RSX-11S system.

HLD — *node/task* — SLD premature disconnect

SLD has disconnected during a checkpoint write before the transfer is complete.

14.4 NCP Error Messages

There are two sets of error messages for NCP: a full set for RSX-11M and RSX-11M-PLUS systems and a subset for RSX-11S systems. There are two groups of messages within the full set: one group is RSX system-specific, and the other is standard for all DECnet systems. The error message descriptions below identify each message as being RSX-specific, RSX-11S, or standard DECnet.

14.4.1 RSX-11M/M-PLUS Error Message Format

The general format for an RSX-11M/M-PLUS NCP error message is:

NCP — [*<component>*.]*<command>* *<diagnostic>*, *<error message>*
[,*<error detail>*] [*<extra text>*]

where

component — is the affected component (for example, LINE *line-id*). Appears only in a command affecting multiple components.

command — is the command that caused the error.

diagnostic — is one of the following values:

FAILED

The command was accepted by NCP, but execution failed.

NOT ACCEPTED

NCP would not accept the command, and execution was not attempted. When NCP rejects a command, it displays the command line with two right angle brackets at the point where NCP stopped parsing. If NCP was parsing a pair of words, it points to the first word of the pair even if the second word was incorrect — for example,

```
SET EXECUTOR >> ROUTING RIMER 50
```

NCP does not recognize ROUTING RIMER. Since both words make up one parameter, NCP rejects the parameter and places brackets in front of it.

error message is the major reason for the failure. In most cases, these reasons are identical for all DECnet implementations (see the *DNA Network Management Functional Specification*).

If NCP cannot interpret an internal error code, the error message field contains one of the following:

Management return #*n*

An error code of *n* (decimal) was returned by network management software.

System return #*n*

An error code of *n* (decimal) was returned by the RSX operating system software.

error detail is a detailed explanation of the failure (for certain error messages). All DECnet implementations have the same definitions for these messages. See the *DNA Network Management Functional Specification* for details.

If NCP cannot interpret an error detail code, the error detail field contains *detail #n*, meaning that an error detail of *n* (decimal) was returned by network management software.

extra text is an additional system-specific explanation of the error condition. For a description of the contents of this field in an error from a non-DECnet-RSX implementation, consult the documentation for that system.

14.4.2 RSX-11S Error Message Format

The general format for an RSX-11S NCP error message is:

NCP — <*error message*>

where

error message See description given above for RSX-11M/M-PLUS.

14.4.3 NCP Error Messages

NCP — Bad loopback response

(Standard DECnet) The message that was returned in a loopback test did not match the message that was sent. This can be due to a loopback protocol violation, bad data returned, or bad message length returned.

NCP — Bad management response

(Standard DECnet) The information returned to NCP either from its internal network management functions or from the network management listener was improperly formatted or contained an invalid value. This error condition implies a programming error in the module that responded to NCP.

NCP — Circuit in wrong state

(RSX-11S) The current operational state of the circuit precludes the requested operation.

NCP — Component in wrong state

(Standard DECnet) The current operational state of the component precludes the requested operation (for example, you cannot down-line load a line that is off). The error detail included with this message identifies the component (for example, circuit, line, node, and so on).

When a LOOP CIRCUIT command is given on an RSX system, one of the following RSX system-specific extra text messages may be included:

Circuit is loopback node

The circuit is currently associated with a loop node name, and its state cannot be changed to do the circuit test.

Circuit not restarted

An attempt was made to return the circuit to normal use before the circuit had been restarted.

Device off line

The circuit is no longer on the I/O bus. This message is specific to RSX-11M-PLUS.

Other management circuit service active

The circuit is currently being used to perform some other network management circuit service function, such as an up-line dump or a down-line load.

Reassign circuit to former owner failed

At the completion of the test, the owner attempted to return the circuit to normal use. The attempt failed because the circuit could not be given back to the owner.

NCP — Error reading command

(RSX-specific) NCP could not read a command from either the terminal or a command file because of a hardware error or an unrecognized command read error.

NCP — Exceeded command file nesting depth

(RSX-specific) A command file attempted to start another command file when there were already six command files nested.

NCP — File I/O error

(Standard DECnet) A hardware error was encountered while reading or writing one of the files necessary to the requested operation. The error detail included with this message identifies the type of file:

Dump file	Output file for an up-line dump
Load file	Input file for a down-line load
Permanent data base	File containing default parameters
Secondary dumper	Input file for the secondary dumper program that runs in the target node
Secondary loader	Input file for the secondary loader program that runs in the target node
Tertiary loader	Input file for the tertiary loader program that runs in the target node

NCP — File open error

(Standard DECnet) One of the files necessary for the requested operation could not be reserved for use. The error detail is the same as that described above for the File I/O error message.

NCP — File privilege violation

(RSX-specific) The file specified with the TO option in a SHOW command is not available to the user of NCP.

NCP — Hardware failure

(Standard DECnet) The requested operation could not be performed because a related hardware component failed to function as needed.

NCP — Incompatible management version

(Standard DECnet) The network management listener version is incompatible with NCP.

NCP — Invalid device name syntax

(RSX-specific) The device name in a command file or TO specification did not have legal syntax.

NCP — Invalid directory syntax

(RSX-specific) The directory identification (UIC) in a command file or a TO specification did not have legal syntax.

NCP — Invalid file contents

(Standard DECnet) The requested operation could not be performed because the files contained data of an invalid form or value. The error detail is the same as that described previously for the File I/O error message.

NCP — Invalid file name syntax

(RSX-specific) The file name in a command file or a TO file specification did not have legal syntax.

NCP — Invalid identification format

(Standard DECnet) The identification of the component involved in the requested operation did not have proper syntax (for example, a numeric character in a node name). The error detail included with this message identifies the type of component (for example, line, node, circuit, and so on).

NCP — Invalid length value, maximum loop length = 128

(RSX-11S) The value of the length parameter in a LOOP NODE command exceeds the buffering capability of subset XPT. Use a smaller value for the length parameter.

NCP — Invalid message format

(Standard DECnet) The information sent by NCP either to its internal network management functions or to a network management listener was improperly formatted or contained an invalid value. This error condition implies a programming error in NCP.

NCP — Invalid parameter grouping

(Standard DECnet) The parameters furnished by the user for the requested operation cannot be included in a single command.

NCP — Invalid parameter value

(Standard DECnet) The value of a parameter furnished by the user for the requested operation was not acceptable (for example, a numeric parameter out of range). The error detail included with this message identifies the type of parameter.

In the case of the events parameter for the LOGGING command, the following RSX system-specific extra text message may be provided:

Event cannot be filtered

The ability to enable or disable one of the specified events is not implemented.

In the case of the length parameter for a LOOP command, one of the following RSX system-specific extra text messages may be included. In each of these cases, the length was more than could be handled, and the maximum length will be included with the error message.

Active looper size exceeded

The requested length exceeds the buffering capability of the active looper task. Reduce the looper's size or reinstall the looper with a memory increment.

Mirror size exceeded

The requested length exceeds the buffering capability of the network management loopback mirror. Either use a smaller length or (on an RSX system) reinstall the mirror with a memory increment.

NCP — Line communication error

(Standard DECnet) The requested operation failed because of communication errors on the involved line. This error condition applies only to functions that involve direct use of a line, such as down-line load and line loop tests.

In the case of a LOOP LINE command, one of the following RSX system-specific extra text messages may be included:

Receiver

The looper failed to receive a message that should have been looped back on the line.

Transmitter

The looper failed to transmit a message on the line. This is specifically a failure of the transmit function.

NCP — Line protocol error

(Standard DECnet) The requested operation failed because of protocol errors on the involved line. This usually implies either incompatible line protocols or protocol-programming errors. There is a possibility that the condition is caused by a line hardware error that was not detected by the line protocol.

Line protocol can mean either the Data Link Protocol or the service operation protocol.

This error condition applies only to functions involving direct use of a line, such as down-line load and line loop tests.

NCP — Listener connect failed

(Standard DECnet) The logical link from NCP to the network management listener could not be connected. This error message generally has one of the following error details:

Access control rejected

The remote node or the network management listener could not understand or would not accept the access control information.

Invalid node name format

The executor rejected the format of the remote node name (for example, the name contained illegal characters or was too long).

Invalid object name format

The remote node did not understand the object name format used by NCP to identify the network management listener.

Local node shut down

The executor node is in the process of shutting down and will not accept logical link connections.

Network resources

Either the local or the remote node had insufficient network resources to connect the logical link.

No response from object

The network management listener did not respond. This could be, for example, due to its responding too slowly or terminating abnormally.

Node unreachable

No path exists to the remote node.

Object too busy

The remote node had insufficient resources available to forward the connect request to the network management listener.

Remote node shut down

The remote node is in the process of shutting down and will not accept logical link connections.

Unrecognized node name

The destination node name does not correspond to any known node address.

Unrecognized object

The remote node does not have a network management listener.

NCP — Listener link disconnected

(Standard DECnet) The logical link from NCP to the network management listener was unexpectedly disconnected. This error message will have one of the following error details:

Abort by management

An operator or program used network management to abort the logical link. NCP recognizes this condition, although DECnet-RSX does not support the ability to cause it.

Abort by object

The network management listener aborted the logical link. This indicates a programming error in the network management listener.

Disconnect by object

The network management listener disconnected the logical link. This indicates a programming error in the network management listener.

Node or object failed

The network aborted the logical link for one of the following reasons: The remote node terminated abnormally, or the network management listener terminated abnormally.

Node unreachable

The network aborted the logical link because it could no longer find a path to the remote node.

NCP — Management program error

(Standard DECnet) The network management software has detected an internal programming error.

NCP — Mirror connect failed

(Standard DECnet) The logical link to the network management loopback mirror could not be connected. This error message generally has one of the following error details:

Access control rejected

The remote node or the network management loopback mirror either could not understand or would not accept the access control information.

Invalid node name format

The executor rejected the format of the remote node name (for example, the name contained illegal characters or was too long).

Invalid object name format

The remote node did not understand the object name format used to identify the network management loopback mirror.

Local node shut down

The executor node is in the process of shutting down and is accepting no more logical link connections.

Network resources

Either the local or the remote node had insufficient network resources to connect the logical link.

No response from object

The network management loopback mirror did not respond. This could be, for example, due to its responding too slowly or terminating abnormally.

Node unreachable

No path exists to the remote node.

Object too busy

The remote node had insufficient resources available to forward the connect request to the network management loopback mirror.

Rejected by object

The logical link could not be connected because the network management loopback mirror rejected the connection. This most likely implies that the loopback mirror is too busy to accept another logical link.

Remote node shut down

The remote node is in the process of shutting down and will accept no more logical link connections.

Unrecognized node name

The destination node name does not correspond to any known node address.

Unrecognized object

The remote node does not have a network management loopback mirror.

NCP — Mirror link disconnected

(Standard DECnet) The logical link from NCP to the network management listener was unexpectedly disconnected. This error message generally has one of the following error details:

Abort by management

An operator or program used network management to abort the logical link. NCP recognizes this condition, although DECnet-RSX does not support the ability to cause it.

Abort by object

The network management loopback mirror aborted the logical link. This indicates a programming error in the network management loopback mirror.

Disconnect by object

The network management loopback mirror disconnected the logical link. This indicates a programming error in the network management loopback mirror.

Node or object failed

The network aborted the logical link because either the remote node or the network management loopback mirror terminated abnormally.

Node unreachable

The network aborted the logical link because it could no longer find a path to the remote node.

NCP — NCP program error

(RSX-specific) NCP detected an internal programming error.

NCP — No room for new entry

(Standard DECnet) The requested operation could not be performed because it required the addition of a new entry in some data base and that data base was full.

NCP — Not remotely executable

(Standard DECnet) The requested operation cannot be sent to a remote node.

NCP — Operation failure

(RSX-11S) The requested operation failed for one of the following reasons: a circuit was not owned by XPT or could not be turned off, a node was not in the XPT data base, or XPT was not loaded.

NCP — Operation failure

(Standard DECnet) The requested operation failed. For some RSX system-specific failures, one of the following extra text messages may be included:

DLX error #*n*

A direct line control (DLX) function failed with decimal error code *n*. Consult DLX documentation for an explanation of the error code.

Insufficient buffer space for additional aliases

The network management software could not internally buffer all the aliases that could have been displayed.

Insufficient buffer space for additional nodes

The network management software could not internally buffer all the nodes that could have been displayed.

Insufficient buffer space for additional objects

The network management software could not internally buffer all the objects that could have been displayed.

Network initializer function failed

The network initializer (NTINIT) failed to perform its part of the requested operation. The network initializer will write an error message to the requesting terminal or, in the case of a remotely executed command, to the system console.

Network initializer not installed

The network initializer could not run because it is not properly installed.

Network loader function failed

The network loader (NTL) failed to perform its part of the requested operation. The network loader will write an error message to the requesting terminal or, in the case of a remotely executed command, to the system console.

Network loader not installed

The network loader could not run because it is not properly installed.

Node in wrong state

The requesting operation could not be performed because the executor node is in the wrong state.

PSI not generated

An attempt was made to perform a PSI operation on a system that does not support PSI.

Server task not installed

The separate task needed to perform the requested operation is not properly installed. This could be the active looper for a loop test, the down-line loader for a down-line load or trigger, or the up-line dumper for an up-line dump.

NCP — Oversized management command response

(Standard DECnet) The command message sent by NCP was too big for the network management listener.

NCP — Oversized management response

(Standard DECnet) The response message returned by the network management listener was too big for NCP.

NCP — Parameter missing

(Standard DECnet) The user did not supply a parameter that is necessary for the requested operation. The error detail included with this message identifies the type of the parameter.

NCP — Parameter not applicable

(Standard DECnet) The user supplied a parameter that is not applicable to the requested operation on the specified component. The error detail included with this message identifies the type of parameter.

NCP — Parameter value too long

(Standard DECnet) The user supplied a parameter value that was too long to be accepted by the network management listener. The error detail included with this message identifies the type of the parameter.

NCP — Privilege violation

(Standard DECnet) The user does not have sufficient privilege to perform the requested operation.

In the case of a LOOP LINE or LOOP CIRCUIT command, the following RSX system-specific extra text message may print out:

Service access denied

The line or circuit is not in SERVICE state and the nonprivileged user cannot perform a line or circuit test.

NCP — Resource error

(Standard DECnet) Network management had insufficient internal resources to perform the requested operation.

NCP — System in wrong state

(RSX-specific) NCP cannot perform the requested function because the DECnet system (the Communications Executive) has not been properly initialized.

NCP — System-specific management function not supported

(Standard DECnet) The requested operation is RSX system specific and is not supported by the network management listener.

NCP — Unrecognized circuit identification

(RSX-11S) The circuit specified by the user does not exist or did not contain necessary information (such as a tributary number).

NCP — Unrecognized command

(Standard DECnet) NCP does not have the command the user entered.

NCP — Unrecognized command

(RSX-11S) NCP does not have the command that the user entered, or the command contained an unrecognized keyword or an invalid parameter value (for example, a node name contained 7 characters instead of the allowed 6-character name).

NCP — Unrecognized component

(Standard DECnet) The component specified by the user does not exist. The error detail included with this message identifies the type of component (for example, line, node, circuit, and so forth).

In the case of the LOOP LINE command, the following RSX system-specific extra text message may be displayed:

Or line cleared

The line may be defined in the system configuration, but it is in the CLEARED state.

NCP — Unrecognized device or unit

(RSX-specific) The device name or unit number in a command file or a TO file specification could not be found by the RSX operating system.

NCP — Unrecognized file, device, or directory

(RSX-specific) The file name, device name, or directory (UIC) in a command file or a TO file specification could not be found by the RSX operating system.

NCP — Unrecognized function or option

(Standard DECnet) The requested operation is not implemented by the executor.

NCP — Unrecognized keyword

(Standard DECnet) One of the keywords in a command is unknown to NCP.

NCP — Unrecognized line identification

(RSX-11S) The line specified by the user does not exist or did not contain necessary information (such as a tributary number).

NCP — Unrecognized node identification

(RSX-11S) The node specified by the user does not exist.

NCP — Unrecognized parameter type

(Standard DECnet) One of the parameters given by the user is not implemented by the executor. The error detail included with this message identifies the type of parameter.

NCP — Unrecognized value

(Standard DECnet) A parameter value given by the user was unknown to NCP.

NCP — Value out of range

(Standard DECnet) A parameter value given by the user is out of the range implemented in NCP.

14.5 NDA Error Messages

NDA prints an error message on your terminal when it detects one of the error conditions described below. These error messages reflect operational conditions and must not be confused with the diagnostic analysis messages that NDA generates during the analysis and prints in the analysis listing.

Note that a few of the messages listed below are informational in nature, while most are fatal errors resulting in termination of the analysis.

NDA — Analysis output must be directed to an explicit device or file

(Fatal) NDA requires an explicit output file specification. There are no default output file names.

NDA — Analysis terminated after *n* pages

(Informational) NDA terminated the analysis after generating *n* pages of analysis output.

NDA — Command I/O error

(Fatal) The system returned an error when NDA attempted to read a command line.

NDA — Command line syntax error

(Fatal) NDA detected an error in the syntax of an NDA command line. NDA points to the error within the command line.

NDA — Crash dump must be input from an explicit device or file

(Fatal) The crash dump input file specification must be explicit. There is no default file specification for the crash dump input.

NDA — Device driver missing

(Fatal) You have not loaded the driver for the crash dump input device.

NDA — Error reading crash dump

(Fatal) NDA either could not read or could not access the crash dump input device. The device may not be properly allocated or may contain bad blocks.

NDA — Error reading file *filename*

(Fatal) The system returned an error when NDA attempted to read the crash dump file. This could be caused by a device error or a bad block on the volume.

NDA — Error reading symbol file *filename*

(Fatal) The system returned an error when NDA attempted to read the symbol table definition file.

NDA — Error writing analysis file

(Fatal) The system returned an error when NDA attempted to write a line into the analysis listing file. This could be caused by a problem with the device, a full volume, or a bad block on the volume.

NDA — Error writing dump file *filename* Error=*n* DSW=*n*

(Fatal) The system returned an error when NDA attempted to write into the binary output file. This condition could be caused by a problem with the device, a full volume, or a bad block on the volume.

NDA — Errors detected: *n*

(Informational) NDA detected *n* analysis errors during the run.

NDA — Failed to assign LUN to input device *ddu*

(Fatal) A directive to assign a LUN failed when NDA attempted to use it to attach the specified input device before reading the crash dump from the device. The directive to assign a LUN fails if the device name in the NDA command line is invalid.

NDA — Failed to extend page buffer — *n* pages available

(Informational) A directive to extend the task failed when NDA attempted to use it to expand the page buffer. This problem causes the analysis to take longer, but the analysis continues with a buffer of *n* pages, each 266 words long.

NDA — Failed to open input file *filename*

(Fatal) One of the following conditions exists:

- The specified device does not exist.
- The volume is not mounted.
- A problem exists with the device.
- The specified UFD does not exist.
- The specified file does not exist.
- You do not have read access privileges.

NDA — Failed to open output file *filename*

(Fatal) One of the following conditions exists:

- The specified device does not exist.
- The volume is not mounted.
- A problem exists with the device.
- The specified UFD does not exist.
- The volume is full or the device is write protected.
- You do not have write access privileges to UFD.

NDA — Illegal switch

(Fatal) You have specified an unknown switch or have used a valid switch after the wrong file specification.

NDA — Illegal trap — snapshot dump being attempted

(Fatal) NDA has aborted after detecting an odd address or some other type of fault. This is an indication of a software problem.

NDA — Indirect command syntax error

(Fatal) The name of the indirect command file (@*filename*) is syntactically incorrect.

NDA — Indirect file open failure

(Fatal) NDA could not open an indirect command file specified as @*filename* in the NDA command line.

NDA — Maximum indirect file depth exceeded

(Fatal) You have exceeded the maximum allowable number of nested indirect command files (only one level of indirection is permitted in NDA). The rest of the top level command file is executed.

NDA — No input file specified

(Fatal) You did not supply an input file specification for the crash dump.

NDA — No output file specified

(Fatal) You have not specified an output file.

NDA — Output dump file name must be explicit

(Fatal) You have not specified an output dump file.

NDA — Premature end of dump input — *filename* being zero filled

(Informational) NDA reached the end of the medium (or end-of-file mark, if a magnetic tape) before the crash dump output file had been completely filled. If you expected the file to be completely filled by the dump, this condition could indicate a problem.

NDA — Symbol file *filename* has illegal format

(Fatal) The specified executive symbol table file has an improper format, probably caused by entry of the wrong file name. However, this message could also indicate a problem with the device or medium on which the file is located.

NDA — Symbol *symbol-name* not defined in symbol file

(Fatal) NDA did not find a symbol it required for the analysis in the specified executive symbol table file. The analysis may be restarted specifying the /SYM switch, forcing NDA to continue with the analysis, using a zero value for all undefined symbols.

NDA — Transfer complete — *ddu* may be unloaded

(Informational) The transfer of the crash dump to the output file is finished; you may unload the crash dump device. This message occurs only when you have specified a binary save file in the command string to NDA.

NDA — Unknown get command line error

(Fatal) Unrecognized error has occurred when NDA attempted to read a command line.

NDA — Warning kernel PARs do not contain proper values

(Informational) This message indicates that the values contained in the page address registers are invalid. NDA attempts to finish the analysis using the indicated values. When restarting the analysis, you must specify the /KMR switch (Section 10.7.2). This switch forces NDA to use standard mapping values when converting kernel virtual addresses to physical memory addresses.

14.6 NTD Error Messages

NTDEMO prints an error message on your terminal if it detects one of the conditions described below when you invoke the NTD utility.

NTDEMO — Network not loaded

The network at the node is not loaded (that is, no network software is running at the local node).

NTDEMO — Unable to connect to remote server

NTD is unable to communicate with NTDEMO on the node you requested (that is, the host node if you typed >NTD, or the node you specified if you typed >NTD *node-id*). This could be due to any one of the following conditions:

- The node is unreachable.
- The node does not support the NTDEMO server. (This software runs only on DECnet-RSX/IAS nodes.)
- The node does not have NTDEMO installed.
- The maximum number of links to NTDEMO are busy. (This value is defined during network generation.)

NTDEMO — Unable to open network

The network at the node is loaded, but not currently available for network operations.

14.7 NTINIT Error Messages

The network initializer (NTINIT) is used with either NCP or VNP to initialize the network software. All NTINIT errors are fatal. The attempted operation cannot succeed unless the problem identified in the message is corrected.

When an NCP or a VNP command executes locally, error messages are displayed at the requesting terminal. When the command executes remotely, error messages are displayed on the remote system console.

Some of the error messages include words printed in italics. The italicized words are variables. NTINIT replaces them with a specific term. For example, in the message NTINIT — Failed to load line *line-id*, the initializer replaces *line-id* with the actual line identification.

NTINIT — Adjacency data base allocation failure

There is not enough free space in the network pool (POOL..) to allocate the adjacency data base. Increase the size of the network pool byte area with CFE and reload the network.

NTINIT — Clearing system

NTINIT is removing the network software from the system because of a failure on the marked-for-load scan. To recover, use CFE to correct the condition that caused the failure. Then execute another SET SYSTEM command and try to load again. If the failure recurs, shut down more lines and circuits before loading again. Repeat this procedure until the load is successful or until all lines and circuits are shut down.

If the error continues after all lines and circuits are shut down, one of the processes is at fault. Set one process to the CLEAR state and try the SET SYSTEM command again. Repeat this action using a different process until the SET SYSTEM command is successful. The last process to be cleared is the faulty one. Then use CFE DEFINE commands to reload the lines, circuits, and processes (except the erring one).

NTINIT — Comm Exec not loaded

The Communications Executive is not loaded in main memory. Use the NCP SET SYSTEM command to load the network software.

NTINIT — Device *controller-id* offline

The CSR address for the device does not exist on the bus. Use CFE to change the CSR value in the configuration file or use the CSR keyword when issuing the NCP SET LINE command. If the line is set to ON during NETGEN, you must use the following NCP commands to clear the line and reload it:

```
NCP>CLEAR LINE line-id ALL
NCP>SET LINE line-id CSR device-id STATE ON
```

NTINIT — Failed to load line *line-id*

A SET LINE request failed. You will also receive an NTL error messsage that gives a more detailed description of the cause of the error (for example, NTL — Device off line).

NTINIT — Failed to load process *process-id*

A SET PROCESS request failed. You will also receive an NTL error message that gives a more detailed description of the cause of the error (for example, NTL — Process not in system).

NTINIT — Illegal function

An attempt to reprogram the network software resulted in a programming error.

NTINIT — Incompatible with RSX-11M system

The initializer was built with the wrong RSX11M.STB file. Rebuild the network software with the correct RSX11M.STB file.

NTINIT — Incompatible with RSX-11S system

The initializer was built with the wrong RSX11S.STB file. Rebuild the network software with the correct RSX11S.STB file.

NTINIT — Network ACP (NETACP) not installed

NETACP must be installed to perform the SET EXECUTOR STATE ON operation. To clear the problem, use the MCR INSTALL NETCAP command to install the network ACP.

NTINIT — Network loader (NTL...) not installed

NTL must be installed to perform the NCP SET SYSTEM operation. To clear the problem, use the MCR INSTALL NTL command to install the network loader.

NTINIT — NS: assignment failure

The initializer could not assign one of its LUNs to the network device. There is no recovery procedure from this failure.

NTINIT — NW: assignment failure

The initializer could not assign one of its LUNs to the PSI network device. There is no recovery procedure from this failure.

NTINIT — PLB allocation failure

There is not enough free space in the dynamic storage region (DSR) to allocate the XPT physical link block data base. You must run a system generation again, allocating enough free space in the DSR.

NTINIT — Router priority data base allocation failure

There is not enough free space in the dynamic storage region (DSR) to allocate the router priority data base.

NTINIT — Routing control task not installed

The routing control task must be installed to perform the SET EXECUTOR STATE ON command. Use the MCR INSTALL <*netuic*> RCP1 command to install the task.

NTINIT — X25 ACP (X25ACP) not installed

The X25ACP must be installed to set module X25-SERVER to ON. Use the MCR INSTALL X25ACP command to install the X25ACP.

14.8 NTL Error Messages

The network loader (NTL) loads and unloads network software in response to NCP commands. When an NCP command executes locally, NTL error messages are displayed on the requesting terminal. When the command executes remotely, NTL error messages are displayed on the remote console terminal.

There are ten types of NTL error messages. The format of each is outlined below. The term *message-text* used in these sample formats is replaced with the actual message in the error message descriptions that follow. The italicized word *var* is replaced with a run-time variable in the actual message. Some of the variables are network processes, device driver names, controller names, line names, partition names, file names, vector values, and RSX device names. Variable fields are ignored in the alphabetization of error messages in this chapter.

NTL — *message-text*

NTL — *var message-text*

NTL — Config File — *message-text*

NTL — *var*: Device Tables — *message-text*

NTL — *var* Image File — *message-text*

NTL — *var* Microcode File — *message-text*

NTL — *var* Process File — *message-text*

NTL — *var* Symbol Name — *message-text*

NTL — *var* Template — *message-text*

NTL — *var* Template File — *message-text*

NOTE

Only recoverable NTL errors are listed in this guide. Any unlisted error message that you receive reflects an unrecoverable error. Call your Digital representative for assistance.

NTL — *dev-id*: allocation failure

There is not enough space to allocate the RSX-11M device data structures (DCB, UCB, and SCB) for one of the network "devices." In this case, device does not refer to hardware. It refers to a software structure to which the user program assigns its LUNs and directs its QIOs.

NTL — CCB allocation failure

There is not enough space for all the CCBs. It is not known how many were successfully allocated.

NTL — CETAB allocation failure

There is not enough dynamic memory to allocate the CETAB data base.

NTL — Channel *var* still active

An attempt was made to issue a CLEAR PROCESS command for an LLC before all its lines were set to OFF. Set all the lines to OFF and repeat the command.

NTL — Circuit not in system

An attempt was made to load a circuit that is not in the system.

NTL — Comm Exec has been deallocated

The space for the Communications Executive has been deallocated.

NTL — Comm Exec incompatible with RSX-11M system

The Communications Executive was built with the wrong RSX11M.STB file.

NTL — Comm Exec not loaded

An attempt was made to perform an operation that requires the Communications Executive (CEX) to be loaded and it is not. Load the CEX with the NCP SET SYSTEM command.

NTL — Config File — Binary buffer overflow

The NTL end-of-task buffers are not large enough to hold the input. Install NTL with an INC option.

NTL — Config File — Device *var* unibus run mask missing

The unibus run mask is missing for the specified device.

NTL — Config File — Event filter control block allocation failure

There is not enough network pool and DSR to allocate an event filter block.

NTL — Config File — Illegal CSR address

The specified CSR address is not a legal value. Use the correct CSR address.

NTL — Config File — LLC$DF process extension too big

One system process has an extension that makes it larger than the allowed 4K maximum.

NTL — Config File — Non-UMR-mapped area is too large

The non-UMR-mapped area specified in the PAR$DF macro is too large for the network pool partition.

NTL — Config File — Not enough contiguous UMRs

Contiguous UMRs are needed to map the network buffer pool. Issue the SET SYSTEM command before you begin fragmenting the UMRs by running other jobs. If this fails, you can use VNP to load the network software into the operating system file instead of using NCP on the in-core image. VNP is not available on RSX-11M-PLUS systems.

NTL — Config File — Object or remote block allocation failure

There is not enough network pool or DSR to allocate the object or remote block. The line from CETAB where the failure occurred is displayed so that you can determine which object or remote block was not allocated.

NTL — Config File — Open failure (-*error* code.)

The system was unable to open the CETAB.MAC file. See the RSX system documentation for the meaning of the RSX I/O error code.

NTL — Config File — Partition *var* busy

The partition name specified for the network pool is busy.

NTL — Config File — Partition *var* not common

The partition specified for the network pool is not a common partition. Remove that partition and re-create it with the system-controlled attribute:

```
MCR>SET/NOMAIN=partition-name
MCR>SET/MAIN=partition-name:aaaa:bbCOM
```

NTL — Config File — Partition *var* not in system

The partition specified for the network pool is not in the system.

NTL — Config File — Partition *var* not system controlled

NTL requires the partition to be system controlled. This error indicates that the partition was created as a task or common partition. Remove that partition and re-create it with the system-controlled attribute:

```
MCR>SET/NOMAIN=partition-name
MCR>SET/MAIN=partition-name:aaaa:bbSYS
```

NTL — Config File — Partition *var* too fragmented

There is not enough contiguous space in the partition to load a process.

NTL — Config File — Partition *var* too small for RDBs/SDBs

This error pertains only to the network buffer pool partition. The network buffer pool is not large enough for the large and small data buffers. Use CFE to reduce the buffer counts and/or sizes or to re-create a larger partition.

NTL — Config File — Read failure (-*error code*.)

The system was unable to read the CETAB.MAC file. See the RSX system documentation for the meaning of the RSX I/O error code.

NTL — Config File — Secondary CSR off line

The specified secondary CSR (from CETAB.MAC) is off line.

NTL — Config File — Sub-PCB allocation failure

There is not enough space for a partition control block for the network buffer pool.

NTL — Config File — SVC descriptor allocation failure

There was not enough DSR or network pool to allocate the SVC descriptor.

NTL — Config File — UMR block allocation failure

There is not enough space for a UMR assignment block.

NTL — Config File — Vector not in system

The controller's vector address specified in the configuration file does not exist in your system's vector space.

NTL — *var* Device I/O error, CSR = *var*, RAM address = *var* was written, *var* was read

An attempt to load microcode for a KMC device failed due to an input/output failure. One possible cause could be a hardware failure.

NTL — Device off line

An attempt was made to load a device that is off line.

NTL — Device *var* off line

An attempt was made to load a device that is off line.

NTL — Device *dev-id*: still active

An attempt was made to clear an LLC process when the associated device is still active.

NTL — Device *dev-id*: still mounted

An attempt was made to clear an LLC process when the device associated with it is still mounted.

NTL — *dev-id*: driver already resident

The executive data structure for the driver associated with the LLC process being loaded already exists.

NTL — *var* ICB allocation failure

There is not enough space in the dynamic storage region (DSR) to allocate an interrupt control block (ICB).

NTL — *var* Image File — Label block read failure (-*error code*.)

The system is unable to read a label block. See the RSX system documentation for the meaning of the RSX I/O error code.

NTL — *var* Image File — Open failure (-*error code*.)

The attempt to open the specified file failed. See the RSX system documentation for the meaning of the RSX I/O error code.

NTL — *var* Image File — Read failure (-*error code*.)

The attempt to read a record from a specified file failed. See the RSX system documentation for the meaning of the RSX I/O error code.

NTL — Incompatible with Comm Exec

NTL has been built with the wrong CEX.STB file. Rebuild NTL with the correct CEX.STB file.

NTL — Incompatible with RSX-11M system

NTL has been built with the wrong RSX11M.STB file. Rebuild NTL with the correct RSX11M.STB file.

NTL — *var* KRB allocation failure

There is not enough space in either the network pool or the dynamic storage region for the controller request block (KRB).

NTL — Line not in system

An attempt was made to set a line that does not exist.

NTL — Line still active

An attempt was made to clear a line that was still ON. Set the line to OFF and reenter the command.

NTL — Line *var* still ready

An attempt was made to perform a CLEAR PROCESS on a DLC or a DDM controller before all of its circuits were turned off and unloaded. Set the circuits to OFF and reenter the command.

NTL — *var* Line table allocation failure

There is not enough space in DSR for either a device line table or an LLC process data base.

NTL — *var* Microcode File — Label block read failure (*error-code.*)

An attempt to read the label block specified in the command failed due to an input/output error. See the RSX system documentation for the meaning of the RSX I/O error code.

NTL — *var* Microcode File — Open failure (*error-code.*)

An attempt to open the file specified in the command failed due to an input/output error. See the RSX system documentation for the meaning of the RSX I/O error code.

NTL — *var* Microcode File — Read failure (*error-code.*)

An attempt to read a record from the specified file failed due to an input/output error. See the RSX system documentation for the meaning of the RSX I/O error code.

NTL — MUX update only

This is a reminder that the vector and the CSR for the device specified in the command line have not been changed. All lines on a multiplexer device are loaded when any line on the device is loaded. Therefore, all subsequent loads for other lines on this device do not change the vector or the CSR.

NTL — *var* Process File — Incompatible with Comm Exec

The process has been built with the wrong CEX.STB file.

NTL — *var* Process File — Label block read failure (*-error-code.*)

An attempt to read a label block for a process task image file failed due to an input/output error. See the RSX system documentation for the meaning of the RSX I/O error code.

NTL — *var* Process File — Open failure (*-error-code.*)

An attempt to open the task image file for the specified process failed due to an input/output error. See the RSX system documentation for the meaning of the RSX I/O error code.

NTL — *var* Process File — Partition *var* busy

The partition is occupied by either another network process or an RSX-11M resident library. The current occupant must be unloaded.

NTL — *var* Process File — Partition *var* not common

NTL requires the partition to be common. The partition has been created as either task or system controlled. Remove the partition and re-create it with the common attribute:

```
MCR>SET/NOMAIN=partition-name
MCR>SET/MAIN=partition-name:aaaa:bbb:COM
```

NTL — *var* Process File — Partition *var* not in Exec space

The Communications Executive (CEX) partition must be contained completely in the first 20K words of physical memory. This error indicates that the ending address of the CEX partition is larger than this limit. The CEX contains an end-of-partition buffer pool. If you exceed the limit by a small amount (100 or 200 octal), you can try to reduce the CEX's partition size. Otherwise, you should repeat the NETGEN.

NTL — *var* Process File — Partition *var* not in system

The partition specified in your command does not exist. Either create the partition that NTL is seeking, or rebuild the component to change the partition assignment.

NTL — *var* Process File — Partition *var* sub-PCB allocation failure

There is not enough space for a partition control block (PCB) for the named process.

NTL — *var* Process File — Partition *var* too fragmented

This error pertains to system-controlled partitions only. The size of the largest free space in the partition is smaller than the size of the process being loaded. The current occupants of the partition must be rearranged to make a free space of the size needed. NTL does not activate the RSX-11M shuffler task. Manual recovery is necessary.

NTL — *var* Process File — Partition *var* too small

This error pertains to the common partition only. The partition is smaller than the size of the process being loaded. The partition must be created with a larger size. Refer to the TKB map for the process to determine the size needed.

NTL — *var* Process File — Read failure (*-error-code.*)

An attempt to read a record from the process task image file failed due to an input/output error. See the RSX system documentation for the meaning of the RSX I/O error code.

NTL — Process may not be unloaded

An attempt was made to unload a process that is required for network operation and cannot be unloaded.

NTL — Process not in system

An attempt was made to set a nonexistent process.

NTL — Reconfiguration device not in system

The reconfiguration device for the network devices is not loaded. Load the device with the DECnet process called DLX.

NTL — *var* Symbol Table File — Open failure (*-error-code.*)

An attempt to open the system table file for the specified process failed due to an input/output error. See the RSX system documentation for the meaning of the RSX I/O error code.

NTL — *var* Symbol Table File — Read failure (*-error-code.*)

An attempt to read a record from the specified system table file failed due to an input/output error. See the RSX system documentation for the meaning of the RSX I/O error code.

NTL — System not built with network support

This message needs no explanation.

NTL — *var* Template — Library block allocation failure

There is no space in the network buffer pool.

NTL — *var* Template — Resource allocation failure

There is not enough space for the allocation size needed. (Another error message follows that contains the allocation type being attempted.)

NTL — *var* Template — Vector *var* not available

The indicated vector is assigned to another user. The vector must be deassigned from the current user.

NTL — *var* Template — Vector *var* not in system

The vector specified in the command exceeds the maximum value assigned at NETGEN.

NTL — *var* Template File — Open failure (*-error-code.*)

An attempt to open the template file for the specified process failed due to an input/output error. See the RSX system documentation for the meaning of the RSX I/O error code.

NTL — *var* Template File — Read failure (*-error-code.*)

An attempt to read a record from the specified template file failed due to an input/output error. See the RSX system documentation for the meaning of the RSX I/O error code.

NTL — Timer cells not found

NTL could not find the process timer cells in the AUX process.

NTL — Vector not in system

The vector specified in the command line does not exist in your system's vector space. The upper limit of the vector space is determined during SYSGEN. Repeat both SYSGEN and NETGEN.

NTL — *var* Vector set-up error

Some other user has stolen one or more of the interrupt vectors. This occurred after NTL first verified that the interrupt vectors were available, but before NTL built the line table and pointed the vector at the table. The vector must be deassigned from the other user.

NTL — * WARNING * — Failed to reinitialize HRC

The network loader was unable to successfully reinitialize HRC after loading a network device. This could affect the use of the system utility ...CON.

NTL — * WARNING * — System name changed to "*name*"

The system name specified during SYSGEN does not match the node name specified during NETGEN. The system name has been changed to the node name in NETGEN and will remain so until the system is rebooted.

14.9 TRI Error Messages

TRI displays an error message on your terminal when it detects one of the errors described below. These messages reflect operational conditions. While most result in termination of the trace interpreter task, a few are for information only.

TRI — All frames were excluded by /LI or /CH

(Informational) There were no frames traced on the lines or channels that you specified with the /LI or /CH switch.

TRI — Cannot open input file

(Fatal) One of the following conditions exists:

- The specified device does not exist.
- The volume is not mounted.
- A problem exists with the device.
- The specified UIC does not exist.
- The specified file does not exist.
- You do not have read access privileges.

TRI — Cannot open output file

(Fatal) One of the following conditions exists:

- The specified device does not exist.
- The volume is not mounted.
- A problem exists with the device.
- The specfied UIC does not exist.
- The volume is full or the device is write protected.
- You do not have write access privileges to the UIC.

TRI — Command I/O error

(Fatal) The system returned an error when TRI attempted to read a command line.

TRI — Error closing output file

(Fatal) TRI failed to close the trace-listing file.

TRI — Error reading input file

(Fatal) The system returned an error when TRI tried to read the trace input file. This could be caused by a device error or a bad block on the volume.

TRI — Error writing output file

(Fatal) The system returned an error when TRI attempted to write a line into the trace-listing file. This could be caused by a problem with the device, a full volume, or a bad block on the volume.

TRI — Failed to spool output file

(Fatal) The system failed to spool the trace-listing file.

TRI — Illegal character in *filespec*

(Fatal) One of your file specifications is incorrect.

TRI — Illegal command string

(Fatal) TRI detected an error in the syntax of the TRI command string.

TRI — Illegal switch or switch value

(Fatal) You have specified an invalid switch in the TRI command string, or you have given an invalid value to a switch.

TRI — Incompatible switches

(Fatal) You have specified an illegal combination of switches. Check that you have not specified any switches other than /SP or /-SP with the /-IN switch.

TRI — Input file is not a trace dump file

(Fatal) The input file you have specified is not a trace dump file produced as output from an NCP SET TRACE command.

TRI — Trace file is empty

(Informational) The trace file specified as the input file is empty.

14.10 VNP Error Messages

VNP — Account field too long

The account field is limited to 16 characters.

VNP — Alias not in system

An attempt was made to reference an alias that is not in the system.

VNP — Attempt to reference block number that is not within the system image file *block-number* BIAS:*n* ADDRESS:*m*

An attempt was made to reference the specified block number, which is not in the system image file. This error is often caused by attempting to top-down load in a system that is greater than 124K or by building a system image file that is smaller than the real memory size while top-down loading. Use CFE to disable top-down loading.

VNP — CETAB data structures are inconsistent

The data structures in the Communications Executive configuration file (CETAB) are inconsistent. Possibly the CETAB.TSK file is corrupt. Perform another NETGEN to obtain an uncorrupt CETAB.TSK file.

VNP — Circuit not in system

The specified circuit is not in the system.

VNP — COMEXEC symbol missing from symbol table *symbol-id*

The specified symbol is not in the Communications Executive symbol table.

VNP — Communications Executive load failed

VNP could not successfully load the Communications Executive.

VNP — Communications Executive symbol table file open failure *error-code filename*

The Communications Executive symbol table file (*filename*) failed to open. The error code is returned by the operating system's file services. All the network files must reside in the UIC defined by the NETUIC in the system image file. Check the NETUIC in the system image file by using the VMR SET /NETUIC command.

VNP — Configuration file read error

VNP could not read a record from the CETAB source file.

VNP — Controller number too large

The controller number on the line identification is illegal.

VNP — Error parsing file name — *filename*

The file name of the operating system disk image file is invalid.

VNP — Event cannot be filtered

This message indicates that the ability to enable or disable one of the specified events is not implemented.

VNP — Executive symbol missing from symbol table *symbol-id*

The specified symbol is not in the Communications Executive symbol table.

VNP — Executive symbol table file open failure *error-code filename*

The RSX executive symbol table file (*filename*) failed to open. The error code is returned by the operating system's file services. The system image file, its symbol table, and all network files must reside under the same UIC.

VNP — Failed to close system image file *error-code*

The system image file failed to close. The error code is returned by the operating system's file services.

VNP — Failed to get command line

VNP could not read a command line.

VNP — Failed to load a process

VNP did not successfully load a process that was marked for load. You can list all processes to find out which process did not load.

VNP — Failed to load a system line

VNP did not successfully load a line that was marked for load. You can list all lines to find out which line did not load.

VNP — Function not implemented

An attempt was made to execute an unimplemented VNP function.

VNP — GEN partition spans highest block allocated

The default partition GEN spans the highest block allocated. Use the PIP command for copying files with the /BL:*n*. switch to enlarge the output file.

VNP — Illegal event code

The event code given is invalid.

VNP — Illegal executive size *nnn*

The executive size (*nnn*) is illegal. This is probably caused by a corrupt system image file.

VNP — Illegal line cost

The line cost must be in the range of 1 to 1022.

VNP — Illegal multipoint active ratio

The multipoint active ratio must be in the range of 1 to 255.

VNP — Illegal multipoint dead ratio

The multipoint dead ratio must be in the range of 1 to 255.

VNP — Illegal node address

The node address must be in the range of 1 to 1023.

VNP — Illegal object type

The object type numbers must be in the range of 1 to 255.

VNP — Illegal tributary address

The tributary addresses must be in the range of 1 to 255.

VNP — Illegal unit CSR address

The unit CSR address is invalid.

VNP — Insufficient space in template-processing buffer (TMP$$A *psect*) Current size is *n* bytes, but it should be expanded to at least *m* bytes

VNP's processing buffer is *n* bytes long, but it requires *m* bytes for the configuration. Edit VNPGEN.CMD and extend the buffer size. Then perform another task build on VNP and try again.

VNP — Invalid hello timer value

The value specified for the hello timer is invalid. The value must be within the range of 1 to 65,535.

VNP — Invalid line identification

The line identification is not in the proper format.

VNP — Invalid node option

Certain node options are valid only on the executor node, while others are not allowed. You must verify that all options selected are valid for the node specified.

VNP — Invalid parameter grouping

The combination of parameters specified is invalid. For example,

```
NCP>SET LOGGING CONSOLE EVENTS 2.0 CIRCUIT DMC-0 NODE A
```

is invalid because you cannot specify two entities to qualify events that are to be logged. You can specify the circuit or the node, but not both. If you want two entities to be logged, you must enter two separate commands.

VNP — I/O error accessing system image file *error-code*

A disk I/O error occurred when VNP accessed the system image file. The error code is returned by the operating system's file services.

VNP — KMC microcode load failed

VNP failed to load the KMC microcode.

VNP — Line name too long

Line names must be 1 to 3 Radix-50 characters.

VNP — Line not assigned to XPT or DLX

An attempt was made to access a line that is not on a network communications device assigned to XPT or DLX.

VNP — Line not in system

The specified line is not in the system.

VNP — Line parameter not applicable

The line option is valid on SET NODE requests only if the node identification is a name.

VNP — Maximum controllers limited to 64

The maximum controller count for the SET PROCESS request is limited to 64.

VNP — Maximum lines limited to 64

The maximum line count for the SET PROCESS request is limited to 64.

VNP — NETACP must be "fixed" in RSX-11S systems

NETACP is not fixed in an RSX-11S System. Fix NETACP using VMR before starting VNP.

VNP — NETACP not installed

NETACP is not installed. Install NETACP using VMR before restarting VNP.

VNP — Network buffer pool is too small

The network buffer pool is not large enough to load the system software. You will have to use VMR on your system image to change the size of the network pool partition.

VNP — Network initializer not installed

The network initializer (NTINIT) is not installed. Install NTINIT with VMR before restarting VNP.

VNP — Network pool partition not in the system

VNP could not find the network pool partition as specified during NETGEN.

VNP — Node in wrong state

The node is not in the proper state to perform the requested function.

VNP — Node name too long

Node names are limited to a maximum of 6 characters.

VNP — Node not in system

An attempt was made to reference a node that is not in the system.

VNP — Non-UMR-mapped area is too large

The non-UMR-mapped portion of the network pool is larger than the entire network pool.

VNP — No part of GEN partition is within system image

The default partition GEN is not within the system image. Use the PIP command for copying files with the /BL:*n*. switch to enlarge the output file.

VNP — Not enough contiguous UMRs to map network pool

The network pool needs too many UMRs to be loaded. You should try to reduce the size of the network pool.

VNP — NTINIT must be "fixed" in RSX-11S systems

NTINIT is not fixed in an RSX-11S system. Fix NTINIT using VMR before restarting VNP.

VNP — Object not in system

An attempt was made to reference an object that is not contained in the system.

VNP — Object/remote block allocation failure

There is not enough network pool or dynamic storage space to allocate a remote name block or an object block.

VNP — Object task name too long

Object task names must be 1 to 6 Radix-50 characters.

VNP — Open failure on configuration file

VNP could not open the CETAB source file.

VNP — Open failure on system image file *error-code filename*

The system image file (*filename*) cannot be opened. The error code is returned by the operating system's file services.

VNP — Owner process name too long

Process names must be 1 to 3 Radix-50 characters.

VNP — Partition name too long

The partition name must be 1 to 6 Radix-50 characters.

VNP — Password field too long

The password field is limited to 8 characters.

VNP — PLB allocation failure

VNP was not able to allocate enough space for the XPT data bases.

VNP — Process name too long

Process names must be 1 to 3 Radix-50 characters.

VNP — Resource error trying to allocate CCBs

There is not enough dynamic storage space in your system to allocate the control buffers.

VNP — Reverse mapping table inconsistent

The reverse mapping table has been corrupted.

VNP — RSX-11M system pool allocation failure

The system pool is empty. Use VMR to increase the size of the system pool.

VNP — Supposed system image file not actually a system image *filename*

The indicated file is not a system image file.

VNP — Syntax error

The command line is not correct.

VNP — System image does not contain a GEN partition

The default partition for network software is GEN. The system image file does not contain a GEN partition.

VNP — System image file corrupted because VNP was terminated abnormally *filename*

A previous execution of VNP terminated abnormally and may have corrupted the system image file (*filename*). Recopy the system image file before restarting VNP.

VNP — System image must be extended to *n* blocks if the GEN partition is to be contained within the image

The default partition GEN is not within the system image. Use the PIP command for copying files with the /BL:*n*. switch to enlarge the output file.

VNP — The Communications Executive data structures are too large for VNP's internal buffer. The buffer (CET$$A *psect*) can be expanded to 2776 bytes maximum (from the current *n* bytes), but this configuration needs *m* bytes. This error is not recoverable without source modifications.

VNP's internal buffers cannot accommodate the Communications Executive configuration file (CETAB). Reduce the size of the configuration.

VNP — Too many object task copies

A maximum of 63 object task copies is allowed.

VNP — Tributary number too large

Tributary numbers must be in the range of 1 to 255.

VNP — UMR block allocation failure

There is not enough dynamic storage space to allocate a UMR block.

VNP — Unable to dynamically allocate network buffer pool

VNP could not allocate enough partition space to load the network pool.

VNP — Unit number too large

The unit number on the line identification is illegal.

VNP — Unspecified error during CONFIG file scan

VNP could not successfully parse all the definitions in the CETAB source file.

VNP — User identification field too long

The user identification field is limited to a maximum of 16 characters.

VNP — XPT channel table inconsistent

The XPT channel table has been corrupted.

VNP — XPT data base not allocated

The XPT process is loaded but the data structures are not.

VNP — XPT process not loaded

The XPT process is not loaded.

PART V
APPENDIXES

- **Appendix A** — Summarizes all CFE, NCP, and VNP commands and their parameters, and identifies those that are RSX system specific.
- **Appendix B** — Lists all valid object type codes with their process types.
- **Appendix C** — Describes the network management logging monitor interface that you can use to enable a user-written program to process network events.
- **Appendix D** — Describes the event message format and lists all event messages by class and type, including a brief description of each.
- **Appendix E** — Lists all counters maintained by the network for lines, circuits, nodes, modules, and the system, and gives a brief description of each.
- **Appendix F** — Lists all network parameter and counter type numbers recognized by DECnet-RSX network management software.

Appendix A
CFE, NCP, and VNP Command Summary

This appendix summarizes the full command sets for the CFE, NCP (RSX-11M/M-PLUS and RSX-11S), and VNP utilities.

Please review the graphic conventions outlined at the front of this manual, especially the usage of braces { } and brackets [ ]. These conventions are used throughout the command summaries to specify parameter selection and optionality. **Shaded text** flags commands or parameters that are valid for PSI users only.

In this appendix only, red ink is used to identify NCP commands and parameters that are RSX system specific. All CFE and VNP commands are RSX system specific.

A.1 CFE Command Summary

All CFE commands are privileged.

```
DEFINE  {CIRCUIT circuit-id  }  [CHANNEL channel-number]
        {KNOWN CIRCUITS      }  [COST cost]
                                [COUNTER TIMER seconds]
                                [DTE dte-address]
                                [HELLO TIMER seconds]
                                [MAXIMUM BROADCAST ROUTERS number]
                                [MAXIMUM DATA byte-count]
                                [MAXIMUM RECALLS retry-count]
                                [MAXIMUM WINDOW block-count]
                                [MULTIPOINT ACTIVE active-ratio]
                                [NUMBER dte-address]
                                [RECALL TIMER seconds]
                                [ROUTER PRIORITY priority]
                                [SERVICE {DISABLE}]
                                [        {ENABLE }]
                                [STATE {OFF}]
                                [      {ON }]
                                [TRIBUTARY trib-address]
                                [USAGE {INCOMING}]
                                [      {OUTGOING}]

DEFINE  EXECUTOR  [ADDRESS node-address]
                  [BROADCAST ROUTING TIMER seconds]
                  [HOST node-address]
                  [IDENTIFICATION id-string]
                  [MAXIMUM ADDRESS node-address]
                  [MAXIMUM BROADCAST NONROUTERS number]
                  [MAXIMUM COST number]
                  [MAXIMUM HOPS number]
                  [MAXIMUM LINKS number]
                  [MAXIMUM NODE COUNTERS number]
                  [NAME node-name]
                  [ROUTING TIMER seconds]
                  [SEGMENT BUFFER SIZE number]
                  [SUBADDRESSES range]
                  [VERIFICATION [STATE] {OFF}]
                  [                     {ON }]
```

```
DEFINE  {LINE line-id    }  [[CONTROLLER] CSR csr-address]
        {KNOWN LINES     }  [CONTROLLER {LOOPBACK}]
                            [           {NORMAL  }]
                            [COUNTER TIMER seconds]
                            [DUPLEX {FULL}]
                            [       {HALF}]
                            [HOLDBACK TIMER milliseconds]
                            [MAXIMUM DATA byte-count]
                            [MAXIMUM RETRANSMITS retry-count]
                            [MAXIMUM WINDOW block-count]
                            [MULTIPOINT DEAD dead-ratio]
                            [PRIORITY hardware-priority]
                            [RETRANSMIT TIMER milliseconds]
                            [SPEED baud-rate]
                            [STATE  {CLEARED}]
                            [       {OFF    }]
                            [       {ON     }]
                            [UNIT CSR csr-address]
                            [VECTOR vector-address]

DEFINE  {KNOWN LOGGING   }  [{EVENTS list  }]
        {LOGGING CONSOLE }  [{KNOWN EVENTS }]
        {LOGGING FILE    }  [STATE {OFF}]
        {LOGGING MONITOR }  [      {ON }]

DEFINE MODULE X25-ACCESS   DESTINATION dest-name   NUMBER dte-address

DEFINE MODULE X25-PROTOCOL  {{DTE dte-address   {[CHANNELS list]               }}
                            {{KNOWN DTES        {[COUNTER TIMER seconds]       }}
                            {                   {[LINE line-id]                }}
                            {                   {[MAXIMUM CIRCUITS count]      }}
                            {                   {[STATE {OFF}]                 }}
                            {                   {[      {ON }]                 }}
                            {{GROUP group-name  {[DTE dte-address]             }}
                            {{KNOWN GROUPS      {[NUMBER group-number]         }}
                            {                   {[TYPE BILATERAL]              }}
                            {[CALL TIMER seconds]                               }
                            {[CLEAR TIMER seconds]                              }
                            {[DEFAULT DATA byte-count]                          }
                            {[DEFAULT WINDOW block-count]                       }
                            {[MAXIMUM DATA byte-count]                          }
                            {[MAXIMUM CLEARS count]                             }
                            {[MAXIMUM RESETS count]                             }
                            {[MAXIMUM RESTARTS count]                           }
                            {[MAXIMUM WINDOW count]                             }
                            {[RESET TIMER seconds]                              }
                            {[RESTART TIMER seconds]                            }
```

```
DEFINE MODULE  {X25-SERVER}  { [COUNTER TIMER seconds]
               {X29-SERVER}  { [MAXIMUM CIRCUITS count]
                             {
                             { {DESTINATION dest-name     [CALL MASK hex-value ] }
                             { {KNOWN DESTINATIONS        [CALL VALUE hex-value] }
                             {                            [GROUP group-name]     }
                             {                            [NUMBER dte-address]   }
                             {                            [OBJECT object-id]     }
                             {                            [PRIORITY priority]    }
                             {                            [SUBADDRESSES range]   }

DEFINE NODE node-id     [DIAGNOSTIC FILE file]
                        [DUMP ADDRESS address]
                        [DUMP COUNT number]
                        [DUMP FILE file]
                        [HARDWARE ADDRESS E-address]
                        [HOST node-id]
                        [LOAD FILE file]
                        [NAME node-name]
                        [SECONDARY LOADER file]
                        [SERVICE CIRCUIT circuit-id]
                        [SERVICE DEVICE device-type]
                        [SERVICE NODE VERSION  {PHASEIII}]
                        [                      {PHASEIV }]
                        [SERVICE PASSWORD password]
                        [TERTIARY LOADER file]

DEFINE OBJECT type-code    [COPIES  {number}]
                           [        {SINGLE}]
                           [NAME object-name]
                           [USER {DEFAULT}]
                           [     {LOGIN  }]
                           [VERIFICATION {INSPECT}]
                           [             {OFF    }]
                           [             {ON     }]

DEFINE  {PROCESS process-name}   [MAXIMUM CONTROLLERS count]
        {KNOWN PROCESSES     }   [MAXIMUM LINES number]
                                 [STATE {CLEARED}]
                                 [      {ON     }]

DEFINE SYSTEM   [LARGE BUFFER SIZE number]
                [LOCATION {FIRSTFIT}]
                [         {TOPDOWN }]
                [MAXIMUM CONTROL BUFFERS number]
                [MAXIMUM LARGE BUFFERS number]
                [MAXIMUM SMALL BUFFERS number]
                [MINIMUM RECEIVE BUFFERS number]
                [POOL BYTE-AREA byte-count]
                [POOL NAME pool-name]
                [POOL PARTITION partition-name]
```

EXIT [PURGE]

HELP [*command*][*component-type*]

KILL

LIST {CIRCUIT *circuit-id*
KNOWN CIRCUITS}

LIST EXECUTOR

LIST {LINE *line-id*
KNOWN LINES}

LIST {KNOWN LOGGING
LOGGING CONSOLE
LOGGING FILE
LOGGING MONITOR}

**LIST MODULE X25-ACCESS {DESTINATION *dest-name*
KNOWN DESTINATIONS}**

**LIST MODULE X25-PROTOCOL [{{DTE *dte-address*
KNOWN DTES}
{GROUP *group-name*
KNOWN GROUPS}}]**

**LIST MODULE {X25-SERVER
X29-SERVER} {DESTINATION *dest-name*
KNOWN DESTINATIONS}**

LIST {NODE *node-id*
KNOWN NODES}

LIST {OBJECT *type-code*
KNOWN OBJECTS}

LIST {PROCESS *process-name*
KNOWN PROCESSES}

LIST SYSTEM

**PURGE {CIRCUIT *circuit-id*
KNOWN CIRCUITS} COUNTER TIMER**

**PURGE {LINE *line-id*
KNOWN LINES} COUNTER TIMER**

$$\text{PURGE} \begin{Bmatrix} \text{KNOWN LOGGING} \\ \text{LOGGING CONSOLE} \\ \text{LOGGING FILE} \\ \text{LOGGING MONITOR} \end{Bmatrix} \begin{Bmatrix} \text{ALL EVENTS} \\ \text{EVENTS } \textit{list} \\ \text{KNOWN EVENTS} \end{Bmatrix}$$

$$\textbf{PURGE MODULE X25-ACCESS} \begin{Bmatrix} \textbf{DESTINATION } \textbf{\textit{dest-name}} \\ \textbf{KNOWN DESTINATIONS} \end{Bmatrix} \textbf{ ALL}$$

$$\textbf{PURGE MODULE X25-PROTOCOL} \left\{ \begin{array}{ll} \begin{Bmatrix} \textbf{DTE } \textbf{\textit{dte-address}} \\ \textbf{KNOWN DTES} \end{Bmatrix} & \begin{Bmatrix} \textbf{ALL} \\ \textbf{COUNTER TIMER} \end{Bmatrix} \\ \begin{Bmatrix} \textbf{GROUP } \textbf{\textit{group-name}} \\ \textbf{KNOWN GROUPS} \end{Bmatrix} & \textbf{ALL} \end{array} \right\}$$

$$\textbf{PURGE MODULE} \begin{Bmatrix} \textbf{X25-SERVER} \\ \textbf{X29-SERVER} \end{Bmatrix} \left\{ \begin{array}{l} \textbf{COUNTER TIMER} \\ \begin{Bmatrix} \textbf{DESTINATION } \textbf{\textit{dest-name}} \\ \textbf{KNOWN DESTINATIONS} \end{Bmatrix} \end{array} \right\} \textbf{ALL}$$

$$\text{PURGE} \begin{Bmatrix} \text{NODE } \textit{node-id} \\ \text{KNOWN NODES} \end{Bmatrix} \begin{array}{l} \text{[ALL]} \\ \text{[DIAGNOSTIC FILE]} \\ \text{[DUMP ADDRESS]} \\ \text{[DUMP COUNT]} \\ \text{[DUMP FILE]} \\ \text{[HARDWARE ADDRESS]} \\ \text{[HOST]} \\ \text{[LOAD FILE]} \\ \text{[SECONDARY [LOADER]]} \\ \text{[SERVICE CIRCUIT]} \\ \text{[SERVICE DEVICE]} \\ \text{[[SERVICE] PASSWORD]} \\ \text{[TERTIARY [LOADER]]} \end{array}$$

$$\text{PURGE} \begin{Bmatrix} \text{OBJECT } \textit{object-type} \\ \text{KNOWN OBJECTS} \end{Bmatrix}$$

A.2 NCP Command Summary (RSX-11M/M-PLUS Systems)

NCP commands and descriptors that are supported by RSX-11S are summarized in a list following this one.

If you specify ALL, you cannot include any other parameters.

* = Command cannot be executed with the TELL prefix.

P = Privileged   NP = Nonprivileged   (SCOPE is always privileged.)

```
P     CLEAR  { ALL ALIASES       } [ [SCOPE] { GLOBAL            } ]
NP           { ALIAS alias-name  }           { TERMINAL term-id  }
NP           { KNOWN ALIASES     }

P     CLEAR  { CIRCUIT circuit-id }  COUNTER TIMER
             { KNOWN CIRCUITS     }

P     CLEAR EXECUTOR   [HOST]
                       [RECEIVE PASSWORD]
                       [TRANSMIT PASSWORD]

NP  * CLEAR EXECUTOR NODE

P     CLEAR  { LINE line-id  }   { ALL           }
             { KNOWN LINES   }   { COUNTER TIMER }

P     CLEAR  { KNOWN LOGGING   }   [NAME]
             { LOGGING CONSOLE }
             { LOGGING FILE    } [ { ALL EVENTS   } [ { CIRCUIT circuit-id              } ] ]
             { LOGGING MONITOR }   { EVENTS list  }   { LINE line-id                    }
                                   { KNOWN EVENTS }   { MODULE { X25-PROTOCOL }         }
                                                      {        { X25-SERVER   }         }
                                                      {        { X29-SERVER   }         }
                                                      { NODE node-id                    }
                                                      [ SINK { EXECUTOR     }           ]
                                                      [      { NODE node-id }           ]

P     CLEAR MODULE X25-ACCESS  { ALL DESTINATIONS      } [ [SCOPE] { GLOBAL           } ]
                               { DESTINATION dest-name }           { TERMINAL term-id }
                               { KNOWN DESTINATIONS    }

P     CLEAR MODULE X25-PROTOCOL  { { DTE dte-address } { ALL           } }
                                 { { KNOWN DTES      } { COUNTER TIMER } }
                                 { { GROUP group-name } ALL               }
                                 { { KNOWN GROUPS     }                   }

P     CLEAR MODULE  { X25-SERVER }  { COUNTER TIMER             }
                    { X29-SERVER }  { { DESTINATION dest-name } }
                                    { { KNOWN DESTINATIONS    } }
```

P CLEAR NODE *node-id* {CIRCUIT | NAME}

P CLEAR {OBJECT *type-code* | KNOWN OBJECTS} ALL

P CLEAR PROCESS *process-name*

P • CLEAR SYSTEM

P CLEAR TRACE {LINE *line-id* | ACTIVE LINES}

NP • EXIT

NP • HELP [*command*][*component-type*]

P KMX-DUMP LINE *line-id* [FILENAME *file*]

P LOAD NODE *node-id*
[ADDRESS *node-address*]
[FROM *file*]
[HOST *node-id*]
[NAME *node-name*]
[PHYSICAL ADDRESS *E-address*]
[SECONDARY [LOADER] *file*]
[SERVICE DEVICE *device-type*]
[SERVICE NODE VERSION {PHASEIII | PHASEIV}]
[[SERVICE] PASSWORD *password*]
[TERTIARY [LOADER] *file*]
[VIA *circuit-id*]

P LOAD VIA *circuit-id*
[ADDRESS *node-address*]
[FROM *file*]
[HOST *node-id*]
[NAME *node-name*]
[PHYSICAL ADDRESS *E-address*]
[SECONDARY [LOADER] *file*]
[SERVICE DEVICE *device-type*]
[SERVICE NODE VERSION {PHASEIII | PHASEIV}]
[[SERVICE] PASSWORD *password*]
[TERTIARY [LOADER] *file*]

```
P    LOOP  { LINE line-id                                                         } [COUNT count]
           { CIRCUIT circuit-id [  HELP {FULL    }                              ] } [LENGTH length]
           {                    [       {RECEIVE }                              ] } [WITH {MIXED }]
           {                    [       {TRANSMIT}                              ] } [     {ONES  }]
           {                    [{[PHYSICAL ADDRESS E-address          ]}       ] } [     {ZEROES}]
           {                    [{[ASSISTANT PHYSICAL ADDRESS E-address]}       ] }
           {                    [{[NODE E-node-name          ]          }       ] }
           {                    [{[ASSISTANT NODE E-node-name]          }       ] }

NP   LOOP  {NODE node-id[acc-con-info]}  [COUNT count]
           {EXECUTOR                  }  [LENGTH length]
                                         [WITH {MIXED }]
                                         [     {ONES  }]
                                         [     {ZEROES}]

NP   SET   ALIAS alias-name   DESTINATION dest-node  [[SCOPE] {GLOBAL          }]
                                                     [        {TERMINAL term-id}]

P    SET   {CIRCUIT circuit-id}  [COST cost]
           {KNOWN CIRCUITS    }  [COUNTER TIMER seconds]
                                 [HELLO TIMER seconds]
                                 [MULTIPOINT ACTIVE active-ratio]
                                 [OWNER {DLX}]
                                 [      {XPT}]
                                 [SERVICE {DISABLE}]
                                 [        {ENABLE }]
                                 [STATE {OFF    }]
                                 [      {ON     }]
                                 [      {SERVICE}]
                                 [TRIBUTARY  trib-address]

P    SET EXECUTOR   [HOST node-id]
                    [RECEIVE PASSWORD password]
                    [ROUTING TIMER seconds]
                    [SEGMENT BUFFER SIZE number]
                  * [STATE {OFF }]
                    [      {ON  }]
                    [      {SHUT}]
                    [SUBADDRESSES range]
                    [TRANSMIT PASSWORD password]
                    [VERIFICATION [STATE] {OFF}]
                    [                     {ON }]

NP   * SET EXECUTOR NODE node-id[acc-con-info]
```

```
P   SET KNOWN LINES    [STATE {OFF}]
                              {ON }

    Loading options:   [ALL]
                       [DEAD TIMER milliseconds]
                       [DELAY TIMER milliseconds]
                       [DUPLEX {FULL}]
                               {HALF}
                       [[LOCATION] {FIRSTFIT}]
                                   {TOPDOWN }

    Loaded options:    [COUNTER TIMER seconds]
                       [OWNER {DLX}]
                              {PLI}

P   SET LINE line-id   [STATE {OFF}]
                              {ON }

    Loading options:   [ALL]
                       [[CONTROLLER] CSR csr-address]
                       [DEAD TIMER milliseconds]
                       [DELAY TIMER milliseconds]
                       [DUPLEX {FULL}]
                               {HALF}
                       [[LOCATION] {FIRSTFIT}]
                                   {TOPDOWN }
                       [MULTIPOINT DEAD dead-ratio]
                       [PRIORITY hardware-priority]
                       [UNIT CSR csr-address]
                       [VECTOR vector-address]

    Loaded options:    [CONTROLLER {LOOPBACK}]
                                   {NORMAL  }
                       [COUNTER TIMER seconds]
                       [OWNER {DLX}]
                              {PLI}

P   SET {KNOWN LOGGING   }  [NAME name                                                      ]
        {LOGGING CONSOLE }  [STATE {OFF}                                                    ]
        {LOGGING FILE    }  [      {ON }                                                    ]
        {LOGGING MONITOR }  [{EVENTS list     [{CIRCUIT circuit-id                     }]}  ]
                            [{KNOWN EVENTS    [{LINE line-id                           }]}  ]
                            [{                [{MODULE {X25-PROTOCOL}                  }]}  ]
                            [{                [{       {X25-SERVER  }                  }]}  ]
                            [{                [{       {X29-SERVER  }                  }]}  ]
                            [{                [{NODE node-id                           }]}  ]
                            [{                [SINK {EXECUTOR    }                      ]}  ]
                            [{                [     {NODE node-id}                      ]}  ]

P   SET MODULE X25-ACCESS  DESTINATION dest-name  NUMBER dte-address -
             [[SCOPE] {GLOBAL          }]
                      {TERMINAL term-id}
```

```
P  SET MODULE X25-PROTOCOL  { {DTE dte-address  }     [COUNTER TIMER seconds]          }
                            { {KNOWN DTES       }     [STATE {OFF  }]                  }
                            {                         [      {ON   }]                  }
                            {                         [      {SHUT }]                  }
                            {                                                          }
                            { {GROUP group-name }     [DTE dte-address      ]          }
                            { {KNOWN GROUPS     }     [NUMBER group-number  ]          }
                            {                         [TYPE BILATERAL]                 }

P  SET MODULE X25-SERVER  { [COUNTER TIMER seconds]                                    }
                          { [STATE {OFF  }]                                            }
                          { [      {ON   }]                                            }
                          { [      {SHUT }]                                            }
                          {                                                            }
                          { DESTINATION dest-name    OBJECT object-id                  }
                          {                          [CALL MASK hex-value ]            }
                          {                          [CALL VALUE hex-value]            }
                          {                          [GROUP group-name]                }
                          {                          [NUMBER dte-address] -            }
                          {                          [PRIORITY priority]               }
                          {                          [SUBADDRESSES range]              }

P  SET MODULE X29-SERVER  { COUNTER TIMER seconds                                      }
                          {                                                            }
                          { DESTINATION dest-name  OBJECT object-id                    }
                          {                        [CALL MASK hex-value ]              }
                          {                        [CALL VALUE hex-value]              }
                          {                        [GROUP group-name]                  }
                          {                        [NUMBER dte-address]                }
                          {                        [PRIORITY priority]                 }
                          {                        [SUBADDRESSES range]                }

P  SET  NODE node-name   CIRCUIT circuit-id

P  SET  NODE node-id   NAME node-name

P  SET OBJECT type-code    [COPIES {number}]
                           [       {SINGLE}]
                           [NAME object-name]
                           [USER {DEFAULT}]
                           [     {LOGIN  }]
                           [VERIFICATION {INSPECT}]
                           [             {OFF    }]
                           [             {ON     }]

P  SET PROCESS process-name   [ALL]
                              [[LOCATION] {FIRSTFIT}]
                              [           {TOPDOWN }]
                              [MAXIMUM CONTROLLERS count]
                              [MAXIMUM LINES count]
                              [PARTITION partition-name]

P  SET SYSTEM

P  SET TRACE  {LINE line-id    }  [BUFFER SIZE block-count]  [STATE {OFF}]
              {ACTIVE LINES    }  [FILE file]                [      {ON }]
```

```
NP   SHOW  { ALL ALIASES       }  [ { CHARACTERISTICS } ]  [ [SCOPE] { GLOBAL           } ]  [TO file]
           { ALIAS alias-name  }  [ { SUMMARY         } ]  [         { TERMINAL term-id } ]
           { KNOWN ALIASES     }

NP   SHOW  { CIRCUIT circuit-id     }  [ { CHARACTERISTICS } ]  [TO file]
           { ACTIVE CIRCUITS        }  [ { COUNTERS        } ]
           { KNOWN CIRCUITS         }  [ { STATUS          } ]
           { SIGNIFICANT CIRCUITS   }  [ { SUMMARY         } ]

NP   SHOW EXECUTOR      [ { CHARACTERISTICS } ]  [TO file]
                        [ { COUNTERS        } ]
                        [ { STATUS          } ]
                        [ { SUMMARY         } ]

NP   SHOW  { LINE line-id         }  [ { CHARACTERISTICS } ]  [TO file]
           { ACTIVE LINES         }  [ { COUNTERS        } ]
           { KNOWN LINES          }  [ { STATUS          } ]
           { SIGNIFICANT LINES    }  [ { SUMMARY         } ]

NP   SHOW  { ACTIVE LOGGING        }  [ { CHARACTERISTICS } ]  [TO file]
           { KNOWN LOGGING         }  [ { EVENTS          } ]  [ { KNOWN SINKS       } ]
           { LOGGING CONSOLE       }  [ { STATUS          } ]  [ { SINK NODE node-id } ]
           { LOGGING FILE          }  [ { SUMMARY         } ]
           { LOGGING MONITOR       }
           { SIGNIFICANT LOGGING   }

NP   SHOW MODULE X25-ACCESS  { ALL DESTINATIONS       }  [ { CHARACTERISTICS } ]  -
                             { DESTINATION dest-name  }  [ { SUMMARY         } ]
                             { KNOWN DESTINATIONS     }

          [ [SCOPE]   { GLOBAL           } ]   [TO file]
          [           { TERMINAL term-id } ]

NP   SHOW MODULE X25-PROTOCOL  [ { DTE dte-address } ]   [ { CHARACTERISTICS } ]  [TO file]
                               [ { KNOWN DTES      } ]   [ { COUNTERS        } ]
                                                         [ { SUMMARY         } ]

                               [ { GROUP group-name } ]  [ { CHARACTERISTICS } ]  [TO file]
                               [ { KNOWN GROUPS     } ]  [ { SUMMARY         } ]

                               [ { CHARACTERISTICS } ]   [TO file]
                               [ { STATUS          } ]
                               [ { SUMMARY         } ]

NP   SHOW MODULE { X25-SERVER }  [ { DESTINATION dest-name } ]  [ { CHARACTERISTICS } ]  [TO file]
                 { X29-SERVER }  [ { KNOWN DESTINATIONS    } ]  [ { SUMMARY         } ]

                                 [ { CHARACTERISTICS } ]  [TO file]
                                 [ { COUNTERS        } ]
                                 [ { STATUS          } ]
                                 [ { SUMMARY         } ]
```

NP SHOW {NODE *node-id* | ACTIVE NODES | ADJACENT NODES | KNOWN NODES | LOOP NODES | SIGNIFICANT NODES} [{CHARACTERISTICS | COUNTERS | STATUS | SUMMARY}] [TO *file*]

NP SHOW {OBJECT *type-code* | KNOWN OBJECTS} [{CHARACTERISTICS | SUMMARY}] [TO *file*]

NP SHOW {PROCESS *process-name* | ACTIVE PROCESSES | KNOWN PROCESSES} [{STATUS | SUMMARY}] [TO *file*]

NP SHOW SYSTEM [{CHARACTERISTICS | COUNTERS | STATUS | SUMMARY}] [TO *file*]

NP **SHOW TRACE [{STATUS | SUMMARY}] [TO *file*]**

NP TELL *node-id*[*acc-con-info*] *ncp-command*

P TRIGGER NODE *node-id* [VIA *circuit-id*]
[PHYSICAL ADDRESS *E-address*]
[[SERVICE] PASSWORD *password*]

P TRIGGER VIA *circuit-id* [PHYSICAL ADDRESS *E-address*]
[[SERVICE] PASSWORD *password*]

P ZERO {CIRCUIT *circuit-id* | KNOWN CIRCUITS} [COUNTERS]

P ZERO EXECUTOR [COUNTERS]

P ZERO {LINE *line-id* | KNOWN LINES} [COUNTERS]

P **ZERO MODULE X25-PROTOCOL {DTE *dte-address* | KNOWN DTES} [COUNTERS]**

P **ZERO MODULE {X25-SERVER | X29-SERVER} [COUNTERS]**

P ZERO {NODE *node-id* | KNOWN NODES} [COUNTERS]

P ZERO SYSTEM [COUNTERS]

A.3 NCP Command Summary (RSX-11S Systems)

The following commands are supported by the RSX-11S NCP and the RSX-11S NICE. The privileged (P) and nonprivileged (NP) classifications apply only to commands sent from a remote system to an RSX-11S NICE that has been built with a privileged password. All commands can be initiated both locally and remotely unless one of the following restrictions is indicated:

† = Command cannot be executed by the RSX-11S NCP.

‡ = Command cannot be initiated from a remote node.

```
NP    LOOP  {NODE node-id[acc-con-info]}  [COUNT count]
            {EXECUTOR                 }  [LENGTH length]
                                         [WITH {MIXED }]
                                         [     {ONES  }]
                                         [     {ZEROES}]

P     SET CIRCUIT circuit-id  STATE {OFF}
                                    {ON }

P     SET EXECUTOR HOST node-id

P   ‡ SET LINE CONTROLLER {LOOPBACK}
                          {NORMAL  }

P     SET LOGGING CONSOLE  STATE {OFF}
                                 {ON }

NP    SHOW {  CIRCUIT circuit-id } [{COUNTERS}]
           {† ACTIVE CIRCUITS    } [{STATUS  }]
           {† KNOWN CIRCUITS     } [{SUMMARY }]

NP    SHOW EXECUTOR [{CHARACTERISTICS}]
                    [{COUNTERS       }]
                    [{STATUS         }]
                    [{SUMMARY        }]

NP    SHOW {  LINE line-id    } [{COUNTERS}]
           {† ACTIVE LINES    } [{STATUS  }]
           {† KNOWN LINES     } [{SUMMARY }]

NP    SHOW LOGGING CONSOLE STATUS

NP    SHOW {  NODE node-id    } [{CHARACTERISTICS}]
           {† ACTIVE NODES    } [{COUNTERS       }]
           {† KNOWN NODES     } [{STATUS         }]
                                [{SUMMARY        }]
```

```
NP    SHOW SYSTEM  [ { CHARACTERISTICS } ]
                   [ { COUNTERS        } ]
                   [ { STATUS          } ]
                   [ { SUMMARY         } ]

P     ZERO  {   CIRCUIT circuit-id }  [COUNTERS]
            {   EXECUTOR           }
            {   LINE line-id       }
            { † NODE node-id       }
            {   SYSTEM             }

P   † ZERO KNOWN { CIRCUITS }  [COUNTERS]
                 { LINES    }
                 { NODES    }
```

A.4 VNP Command Summary

All VNP commands are privileged.

If you specify ALL, you cannot include any other parameters.

```
CLEAR {ALL ALIASES        } [[SCOPE] {GLOBAL          }]
      {ALIAS alias-name   }          {TERMINAL term-id }
      {KNOWN ALIASES      }

CLEAR EXECUTOR [HOST]
               [RECEIVE PASSWORD]
               [TRANSMIT PASSWORD]

CLEAR {LINE line-id } ALL
      {KNOWN LINES  }

CLEAR {KNOWN LOGGING     }  [NAME]
      {LOGGING CONSOLE   }
      {LOGGING FILE      }  [{ALL EVENTS   [{LINE line-id                 }]}]
      {LOGGING MONITOR   }  [{EVENTS list  [{NODE node-id                 }]}]
                            [{KNOWN EVENTS [{SINK {EXECUTOR              }}]}]
                            [{             [{     {NODE {node-id}        }}]}]
                            [{             [{     {     {$HOST  }        }}]}]

CLEAR NODE node-id {CIRCUIT}
                   {NAME   }

CLEAR {OBJECT type-code } [ALL]
      {KNOWN OBJECTS    }

CLEAR PROCESS process-name

CLEAR SYSTEM

EXIT

HELP [command][component-type]

SET  ALIAS alias-name  DESTINATION dest-node [[SCOPE] {GLOBAL          }]
                                                      {TERMINAL term-id }

SET {CIRCUIT circuit-id } [COST cost]
    {KNOWN CIRCUITS     } [HELLO TIMER seconds]
                          [MULTIPOINT ACTIVE active-ratio]
                          [OWNER {DLX}]
                          [      {XPT}]
                          [SERVICE {DISABLE}]
                          [        {ENABLE }]
                          [STATE {OFF    }]
                          [      {ON     }]
                          [      {SERVICE}]
                          [TRIBUTARY trib-address]
```

```
SET EXECUTOR   [HOST node-id]
               [RECEIVE PASSWORD password]
               [ROUTING TIMER seconds]
               [SEGMENT BUFFER SIZE number]
               [STATE {OFF                        }]
               [      {ON  [{FIXED  }]            }]
               [      {    [{UNFIXED}]            }]
               [TRANSMIT PASSWORD password]
               [VERIFICATION [STATE] {OFF}]
               [                     {ON }]

SET KNOWN LINES

Loading options:  [ALL]
                  [DEAD TIMER milliseconds]
                  [DELAY TIMER milliseconds]
                  [DUPLEX {FULL}]
                  [       {HALF}]
                  [[LOCATION] FIRSTFIT]

SET LINE line-id

Loading options:  [ALL]
                  [[CONTROLLER] CSR csr-address]
                  [DEAD TIMER milliseconds]
                  [DELAY TIMER milliseconds]
                  [DUPLEX {FULL}]
                  [       {HALF}]
                  [[LOCATION] FIRSTFIT]
                  [MULTIPOINT DEAD dead-ratio]
                  [PRIORITY hardware-priority]
                  [UNIT CSR csr-address]
                  [VECTOR vector-address]

SET {KNOWN LOGGING  } [ NAME name                                                  ]
    {LOGGING CONSOLE} [ STATE {OFF}                                                ]
    {LOGGING FILE   } [       {ON }                                                ]
    {LOGGING MONITOR} [ {EVENTS list    [{LINE line-id}                        ] } ]
                      [ {KNOWN EVENTS   [{NODE node-id}                        ] } ]
                      [ {               [SINK {EXECUTOR                 }      ] } ]
                      [ {               [     {NODE {node-id}           }      ] } ]
                      [ {               [     {     {$HOST  }           }      ] } ]
```

SET MODULE X25-SERVER STATE {OFF / ON}

```
SET    NODE node-name  CIRCUIT circuit-id

SET    NODE node-id  NAME node-name
```

```
SET OBJECT type-code    [COPIES {number}]
                        [       {SINGLE}]
                        [NAME object-name]
                        [USER {DEFAULT}]
                        [     {LOGIN  }]
                        [VERIFICATION {INSPECT}]
                        [             {OFF    }]
                        [             {ON     }]

SET PROCESS process-name  [ALL]
                          [[LOCATION] {FIRSTFIT}]
                          [           {TOPDOWN }]
                          [MAXIMUM CONTROLLERS count]
                          [MAXIMUM LINES count]
                          [PARTITION partition-name]

SET SYSTEM [TOP]

SHOW {ALL ALIASES      }  [{CHARACTERISTICS}]  [[SCOPE] {GLOBAL          }]
     {ALIAS alias-name }  [{SUMMARY        }]  [        {TERMINAL term-id}]
     {KNOWN ALIASES    }

SHOW {CIRCUIT circuit-id}  [{CHARACTERISTICS}]
     {KNOWN CIRCUITS    }  [{SUMMARY        }]

SHOW  EXECUTOR  [{CHARACTERISTICS}]
                [{SUMMARY        }]

SHOW {LINE line-id  }  [{CHARACTERISTICS}]
     {KNOWN LINES   }  [{SUMMARY        }]

SHOW {KNOWN LOGGING   }  [{CHARACTERISTICS}]  [{KNOWN SINKS        }]
     {LOGGING CONSOLE }  [{EVENTS         }]  [{SINK NODE node-id  }]
     {LOGGING FILE    }  [{SUMMARY        }]
     {LOGGING MONITOR }

SHOW  MODULE X25-ACCESS  {ALL DESTINATIONS     }[{CHARACTERISTICS}] -
                         {DESTINATION dest-name}[{SUMMARY        }]
                         {KNOWN DESTINATIONS   }

            [[SCOPE] {GLOBAL          }]
            [        {TERMINAL term-id}]

SHOW MODULE X25-PROTOCOL  [{{DTE dte-address}    }]  [{CHARACTERISTICS}]
                          [{{KNOWN DTES     }    }]  [{SUMMARY        }]
                          [{                     }]
                          [{{GROUP group-name}   }]
                          [{{KNOWN GROUPS    }   }]

SHOW MODULE {X25-SERVER}  [{DESTINATION dest-name}]  [{CHARACTERISTICS}]
            {X29-SERVER}  [{KNOWN DESTINATIONS   }]  [{SUMMARY        }]
```

$$\text{SHOW} \left\{ \begin{array}{l} \text{NODE } \textit{node-id} \\ \text{ADJACENT NODES} \\ \text{KNOWN NODES} \\ \text{LOOP NODES} \end{array} \right\} \left[\left\{ \begin{array}{l} \text{CHARACTERISTICS} \\ \text{SUMMARY} \end{array} \right\} \right]$$

$$\text{SHOW} \left\{ \begin{array}{l} \text{OBJECT } \textit{type-code} \\ \text{KNOWN OBJECTS} \end{array} \right\} \left[\left\{ \begin{array}{l} \text{CHARACTERISTICS} \\ \text{SUMMARY} \end{array} \right\} \right]$$

$$\text{SHOW} \left\{ \begin{array}{l} \text{PROCESS } \textit{process-name} \\ \text{KNOWN PROCESSES} \end{array} \right\} \text{ [SUMMARY]}$$

$$\text{SHOW SYSTEM} \left[\left\{ \begin{array}{l} \text{CHARACTERISTICS} \\ \text{SUMMARY} \end{array} \right\} \right]$$

Appendix B
Object Type Codes

Table B-1 defines valid object type code values and describes their process type for network management. The values are expressed as decimal byte values. Digital reserves the right to add object types or to make changes to the descriptor formats used by the object types.

Table B-1: Object Type Codes

Object Type Code	Process Type
0	General task, user program
1	File Access Listener — FAL/DAP, Version 1
2-4	Reserved for DECnet use
5	RSX-11M Task Control — Version 1
6-14	Reserved for DECnet use
15	RSX-11M Task Control — Version 2
16	TLK utility
17	File Access Listener — FAL/DAP, Version 4
18	Host Task Loader
19	Network Information and Control Exchange
20	RSTS/E media transfer program
21-22	Reserved for DECnet use
23	Network terminal handler
24	Reserved for DECnet use
25	Network management loopback mirror
26	Network management event receiver
27-41	Reserved for DECnet use
42	Heterogeneous terminal host
43-62	Reserved for DECnet use
63	RSX DECnet test tool
64-127	Reserved for DECnet use
128-255	Reserved for customer extensions

Appendix C
Network Management Event Logger Interface

The logging monitor interface from the DECnet event-logging facility provides a mechanism by which a user-written program can process network events. You must specify the name of the monitoring program and the events to be logged by using the NCP SET LOGGING command, as follows:

NCP>SET LOGGING MONITOR NAME *name* STATE ON
NCP>SET LOGGING MONITOR EVENTS *event-list*

.
.
.

where

name is the name of the program to receive the event information (default: MON...).

event-list identifies one or more event classes and types to be logged. See Chapter 2 for the event list format and see Appendix D for a list of event classes and types.

NOTE

The event monitor facility can be used in conjunction with event logging on the console or to a file.

If the monitor is inactive, it is invoked when the first event is about to be processed. Your program should open the network (OPN$) and use the GND$ directive to retrieve the event information from the mailbox. The mailbox is a message buffer that receives the event information. When events occur, the I/O status block for the GND$ call contains the following on return from the call:

Word 0: Byte 0 = IS.SUC/IE.DAO
Byte 1 = NT.EVT(6)

Word 1: Number of bytes of event information

The mailbox returned by GND$ contains the event information in the following format:

Word 0,1: is the source program name in Radix-50. The source program is the program that sent the event information. EVC... is the name of the program used for locally generated events; EVR... is the name of the program used for remotely generated events.

Word 2-*n*: is the event information given as a NICE protocol message. The format of the information can be found in the *DNA Network Management Functional Specification.*

After processing the event data, the program can keep the mailbox open and wait for further events, or it can exit and be reinvoked when the next event occurs.

Appendix D
Event Class and Type Summary

Events are summarized by class and type in this appendix. In general, event classes relate to specific layers of the DECnet architecture. Event types relate to specific events within an event class.

NOTE

Asterisks are used throughout this appendix to flag events and event classes that are not logged by DECnet-RSX.

D.1 Event Classes

Event logging supports the event classes listed in Table D-1. DECnet-RSX does not log events for classes marked with an asterisk. However, processed events in these classes from other remote nodes are logged.

Table D-1: Event Classes

Event Class	Description
0	Network Management layer
1 *	Application layer
2	Session Control layer
3	End Communications layer
4	Transport layer
5	Data Link layer
6 *	Physical Link layer
7-31 *	Reserved for other common classes
32-63 *	Reserved for RSTS systems
64-95	RSX system specific
96-127 *	Reserved for TOPS-20 systems
128-159 *	Reserved for VMS systems
160-479 *	Reserved for future use

D.2 Event Message Format

Event messages have the following format:

Event type *class.type* [*event-text*]
Occurred *dd-mon-yy hh:mm:ss* on node *address* [(*node-name*)]
[*entity-type* [*entity-name*]]
[*data*]

where

class	is the layer in which the event occurred.
type	is the specific type of event for that class.
event-text	is the text describing the event (does not print for RSX-11S).
address	is the address of the node at which the event occurred.
node-name	is the name of the node at which the event occurred.
dd-mon-yy	is the date (day, month, and year) on which the event occurred.
hh:mm:ss	is the time (hour, minutes, and seconds) at which the event occurred.
entity-type	is one of four types: line, circuit, node, or module.
entity-name	is the name of the entity that caused the event.
data	is event-dependent text that gives more information about the event. Often this text includes the component type and name for which the event applies. It may also provide additional information about the cause of the event.

Example:

```
 Event type 0.3, Automatic service
Occurred 21-OCT-83 08:17:11 on node 19 (PITSBG)
Circuit UNA-0
Service type = Load
Status = Operation failure
```

The following sections list event messages by class and type for each layer. Individual events and entire event classes that are marked with an asterisk are not logged by DECnet-RSX components.

Event messages shown in this appendix have the following format:

0.0 Event records lost

Event class — Event type — Event message text

D.3 Network Management Layer Events

0.0 Event records lost

Events occurred too rapidly for the event logger to buffer them.

This message does not display any event qualifiers.

0.1 Automatic node counters *

A node counter timer expired.

This message displays the address and name of the node to which the event applies and the counters for that node.

0.2 Automatic line counters

A line counter timer expired.

This message displays the name of the line to which the event applies and the counters for that line.

0.3 Automatic service

An adjacent node requested an automatic circuit service operation.

This message displays the name of the circuit to which the event applies, as well as two event qualifiers: the service function performed (load or dump) and the status of the operation (requested, successful, or failed). If the operation fails, this status includes an error message from line watcher (LIN$$$) and details.

0.4 Line counters zeroed *

Line counters were zeroed.

This message displays the name of the line to which the event applies. The event logger records these counters prior to the execution of a request to zero them.

0.5 Node counters zeroed *

Node counters were zeroed.

This message displays the address and name of the node to which the event applies. The message can also include the node counters that were affected. The event logger records these counters prior to the execution of a request to zero them.

0.6 Passive loopback

The software initiated or terminated a passive loopback test on behalf of an adjacent node.

This message displays the circuit name to which the event applies and the state of operation qualifier (initiated or terminated).

0.7 Aborted service request

An adjacent node requested a service over a circuit connected to the local node. However, a problem prevented it from being processed locally.

This message displays the name of the circuit to which the event applies and the reason for failure. Possible reasons are:

Circuit open error

LIN$$$ received a MOP message and was unable to acquire control of the circuit. Possible causes are: LIN$$$ did not have the privilege to perform the operation, or it could not set the substate of the circuit, or the circuit had another owner.

Circuit state change by higher level

The circuit was preempted by a higher priority function. For example, you used NCP to turn the line off.

Receive error

A line error occurred while trying to receive the request.

Receive timeout

The line message receive timer expired before the request could be received from the adjacent node. Possible causes are: the timer was too short, the line error level was too great for any message to get through, or the adjacent node stopped requesting.

Unrecognized request

A message was received but was not recognizable as a request for up-line dumping, down-line loading, or passive loopback testing. The adjacent node may be running an incompatible version of the line service protocol.

0.8 Automatic counters *

A counter timer for something other than a node or line expired (for example, a DTE counter timer).

0.9 Counters zeroed *

Counters for something other than a node or line were zeroed (for example, a DTE counter timer).

D.4 Session Control Layer Events

2.0 Local node state change

The operational state of the local node changed due to an operator command. Note that the transition from SHUT to OFF also happens automatically when the last logical link is disconnected (under normal operation).

This message displays the reason for the state change (operator command or normal operation), the old state (ON, OFF, SHUT, or RESTRICTED), and the new state (ON, OFF, SHUT, or RESTRICTED).

2.1 Access control failure

The local node rejected a connection request due to invalid access control information.

This message displays the name and address of the source node, the object type number and process ID of the source process requesting the connection, the object type number and process ID of the destination process to receive the connection request, and the invalid access control information (user ID, password, and account information).

D.5 End Communications Layer Events

3.0 Invalid message *

ECL received a message that could not be interpreted. This may indicate a software malfunction in either the local or the remote ECL.

This message displays the invalid ECL message. See the *DNA Network Services Protocol Functional Specification* for a description of ECL messages.

3.1 Invalid flow control *

The remote ECL attempted to invalidly modify the local flow control value. This may indicate a software malfunction in either the local or the remote ECL.

This message displays the invalid ECL message and the current flow control value. See the *DNA Network Services Protocol Functional Specification* for a description of flow control.

3.2 Node data base reused

The local node received a connection request from or tried to initiate an outgoing connect to a node for which there is no counter block. All counter blocks have been previously used, and one of the previously used blocks is available for this new node. This results in the loss of node counters for the node that formerly occupied the data base entry.

This message displays the address and name of the node for which the data base entry was formerly used and the counters for that node.

D.6 Transport Layer Events

4.0 Aged packet loss

A packet has been discarded because it has visited too many nodes. This can be a normal occurrence when the network is reconfiguring its routing data bases, or it can be a failure when the value specified for maximum hops is too small. This can cause the value specified for maximum visits to be too small for a path that should be usable.

This message displays the packet header only. This is information from the beginning of the packet. It consists of a hexadecimal byte of flags, the decimal destination and source node addresses, and a hexadecimal byte of forwarding data. See the *DNA Routing Layer Functional Specification* for additional information.

4.1 Node unreachable packet loss

A packet has been discarded because the local node found that the destination node was unreachable. This event provides a trace of what has happened to packets that are not reaching their destinations.

This message displays the packet header and the name of the circuit to which the event applies. The packet header is described in event 4.0.

4.2 Node out of range packet loss

A packet has been discarded because the destination node number was greater than the maximum node number known to the local node. Normally, this results from the addition of a new node to the network without increasing the value specified for maximum address on the local node and still expecting the local node to route packets to the new node.

This message displays the packet header and the name of the circuit to which the event applies. The packet header is described in event 4.0.

4.3 Oversized packet loss

A packet has been discarded because it was too large to forward to the appropriate adjacent node. Normally, this condition occurs because the adjacent node's buffer is too small or the source node sent a packet that was too large. The latter condition can be handled by setting a smaller segment size at the source node.

This message displays the packet header and the name of the circuit over which the packet was to be forwarded. The packet header is described in event 4.0.

4.4 Packet format error

A packet has been discarded due to a format error in the packet header. Usually, this results from a programming error in packet formatting by the adjacent node. It could also result from a circuit error not detected by circuit protocol.

This message displays a packet header and the name of the circuit to which the event applies. This consists of the first 6 bytes of the packet displayed in hexadecimal.

4.5 Partial routing update loss

A routing message containing node addresses greater than the maximum address known to the local node has been received. Subsequently, information on these nodes was lost. This occurs when the value specified for the maximum address on an adjacent node is increased but the same value for the local node is not increased.

This message displays the name of the circuit over which this message was received, the packet header (see event 4.0), and the highest node address in the routing update that was lost.

4.6 Verification reject

Initialization with an adjacent node has failed. The local node received an invalid password. The receive password does not match the remote node's transmit password. You must either change one of the passwords or clear the receive password.

This message displays the name of the circuit to which the event applies and the address and name of the adjacent node that failed to initialize.

4.7 Circuit down — circuit fault

Line and/or device hardware failure.

This message displays the name of the circuit to which the event applies and the reason for the event. Possible reasons are:

Adjacent node address change

The adjacent node changed addresses without going through the normal initialization sequence. This is logged when an adjacent node attempts to initialize with the local node but the adjacent node's address is not in the data base.

Adjacent node address out of range

The adjacent node's address is greater than the maximum address defined for the local node. This can be caused by an incorrectly defined node address or by a failure to increase the maximum node address in the local node's data base when a new node was added.

Adjacent node block size too small

The line block size provided by the adjacent node is too small for normal network operation. The block size may be set incorrectly at the adjacent node.

Adjacent node listener receive timeout

The node has received no message over the data link within the last 30 seconds. Most likely the remote node is not running.

Adjacent node listener received invalid data

A test message sent by the adjacent node contained invalid or corrupted data. This is most likely caused by a hardware problem.

Circuit synchronization lost

The normal circuit protocol was restarted or terminated by the adjacent node. Either a circuit exceeded an error threshold or network management initiated a circuit state change.

Data errors

The circuit was declared down by the local node's circuit protocol handler when the circuit exceeded an error threshold.

Invalid verification seed value

A transport initialization message sent by an adjacent node is improperly formatted. This is most likely caused by a remote network software problem.

Routing update checksum error

A routing update packet failed its internal integrity test

Unexpected packet type

A packet was received out of the normal protocol sequence. For example, the local node received a normal data packet when it expected a verification packet.

Verification receive timeout

A required verification packet was not received from the adjacent node within the required response time. Either packets were lost on the circuit or a failure occurred at the adjacent node.

Version skew

The routing version of the adjacent node is unacceptable to the local node. The operator may have installed incorrect software at the adjacent node.

4.8 Circuit down

A software error occurred on the circuit and the circuit has been taken out of service. Common causes of this event include the following:

- There is a hardware problem with the line and/or a device.
- The remote circuit has recycled. This could result from an incorrect password sent from this node during initialization.
- The operator turned the remote circuit off and on.

This message displays the name of the circuit to which the event applies, the packet header (see description under event 4.0), and the reason for the event (see the reasons listed for event 4.7).

4.9 Circuit down - operator initiated

An operator has turned the circuit off.

This message displays the name of the circuit to which the event applies, the packet header (see description under event 4.0), the expected node address and name, and the reason for the event (see the reasons listed for event 4.7).

4.10 Circuit up

A remote node has been initialized on one of the circuits connected to the local node.

This message displays the name of the circuit to which the event applies, as well as the name and address of the newly initialized (adjacent) node.

NOTE

On a routing node, this event does not imply that the node is reachable. Reachability is determined by the higher level routing algorithms. For the UNA and QNA, the event signifies that all basic initialization has been completed by the device and transceiver.

4.11 Initialization failure — line fault

A remote node failed to initialize with the local node due to a line error.

This message displays the name of the circuit to which the event applies and the reason for the event (see the reasons listed for event 4.7).

4.12 Initialization failure — software fault

A remote node failed to initialize with the local node due to a software error.

This message displays the name of the circuit to which the event applies, the packet header (see description under event 4.0),and the reason for the event (see the reasons listed for event 4.7).

4.13 Initialization failure — operator fault

A remote node failed to initialize with the local node due to an operator error.

This message displays the name of the circuit to which the event applies, the packet header (see description under event 4.0), the reason for the event (see the reasons listed for event 4.7), and the version received from the adjacent node.

4.14 Node reachability change *

There has been a change in the reachability of a particular node.

This message displays the new node status.

4.15 Adjacency up

For broadcast circuits (UNA and QNA), initialization has occurred with another node on the Ethernet. End nodes log this message for only one node.

This message displays the adjacent node number.

4.16 Adjacency rejected

For broadcast circuits (UNA and QNA), initialization has been rejected. The number of broadcast end nodes may have been exceeded.

The message displays the reason for the rejection.

4.18 Adjacency down

The remote node has recycled. This event could result from a remote node restart or an invalid protocol message.

The message displays the reason why the adjacent node is down.

4.19 Adjacency down — operator initiated *

The operator has turned the Ethernet circuit off. For DECnet-RSX, this event is logged as event 4.9 (see above).

D.7 Data Link Layer Events

5.0 Locally initiated state change *

An operator command changed the circuit state.

This message displays the name of the circuit to which the event applies, the old DDCMP state (HALTED, ISTRT, ASTRT, RUNNING, or MAINTENANCE), and the new DDCMP state (see the old DDCMP states). See the *DNA DDCMP Functional Specification* for a description of these states.

5.1 Remotely initiated state change *

A remote user changed the circuit state.

This message displays the name of the circuit to which the event applies and the old and new DDCMP state (see event 5.0 for the states).

5.2 Protocol restart received in maintenance mode *

The remote node restarted normal operation while the local node had the circuit in maintenance mode.

This message displays the name of the circuit to which the event applies.

5.3 Send error threshold *

Too many data transmission errors occurred.

This message displays the name of the circuit to which the event applies.

5.4 Receive error threshold *

Too many data reception errors occurred.

This message displays the name of the circuit to which the event applies.

5.5 Select error threshold *

Too many selection errors occurred.

This message displays the name of the circuit to which the event applies.

5.6 Block header format error *

DDCMP received an invalid block header.

This message displays the name of the circuit to which the event applies.

5.7 Selection address error *

The wrong tributary responded to the polling process. Only multipoint control stations experience this event. This event occurs when one of these stations receives a message that does not match the address of the currently selected tributary.

This message displays the name of the circuit to which the event applies.

5.8 Streaming tributary *

A tributary on the circuit is impeding the use of that circuit.

This message displays the name of the circuit to which the event applies.

5.9 Local buffer too small *

A local buffer is too small to receive a block of data.

This message displays the name of the circuit to which the event applies.

5.10 Restart *

The DTE completed a restart procedure. That is, X.25 level 3 either has sent a restart confirm signal to a received restart command or has received a restart confirm signal to a transmitted restart command.

This message displays the module name, the DTE name, the cause, and a diagnostic.

5.11 State change *

The operational state of a module was changed by the operator command.

This message displays the name of the module whose state was changed, the name of the DTE where the module resides, the reason for the change, the old state, and the new state.

5.12 Retransmit maximum exceeded *

The retry count for the restart retransmission expired.

This message displays the name of the module whose retransmit maximum was exceeded, the name of the DTE where the module resides, and the parameter for which the maximum was set.

5.13 Initialization failure

The line could not be initialized because of a device error.

This message displays the name of the device that could not be enabled. Depending upon the implementation, a reason for the failure might also be reported.

5.14 Send failed

An error occurred during a transmit operation.

This message displays the name of the line over which the transmit operation was attempted and the reason for failure. Possible reasons are:

Carrier check failed

The data link did not sense the receive signal that must accompany a transmit message. There is a failure in either the transmitting or the receiving hardware (possibly with the transceiver or its cable).

Excessive collisions

The maximum number of retransmissions due to collisions has been exceeded. This is caused when too many systems on the Ethernet are trying to transmit at once.

Frame too long *

The transmit message was truncated. Either the local node tried to send a message that exceeded the maximum allowable size or an error in the local hardware cut off transmission before the actual maximum size was reached.

Open circuit

There is a break in the Ethernet coaxial cable. The event message will include an estimated distance to the failure (in bit times). If other systems are reporting the same problem, the failure is on the network cable rather than with local hardware.

Remote failure to defer

A remote system began transmitting after the allowable time for collisions had elapsed. This could indicate a weak transmitter on the local node or a problem with the remote carrier sense circuitry.

Short circuit

Either there is a short circuit in the Ethernet coaxial cable, or the transceiver or transceiver/controller cable failed. The event message will include an estimated distance to the failure (in bit times). If other systems are reporting the same problem, the failure is on the network rather than with local hardware.

5.15 Receive failed

An error occurred during a receive operation.

This message displays the name of the line over which the receive operation was attempted and the reason for failure. Possible reasons are:

Block check error

The message failed the cyclic redundancy check (CRC). Possible causes are electromagnetic interference, late collisions, or improperly set hardware parameters (such as receiver squelch).

Data overrun

The message was lost due to a hardware failure such as insufficient hardware buffers or insufficient CPU time.

Frame too long

The message was discarded because a remote system sent a message that exceeded the maximum allowable length.

Framing error

The message did not contain a multiple of 8-bit bytes. Possible causes are electromagnetic interference, late collisions, or improperly set hardware parameters (such as receiver squelch).

System buffer unavailable

The message was discarded because there was no system buffer available to receive it. There are not enough system buffers on the local system.

Unrecognized frame destination *

The message was discarded because there was no active user with the specified protocol type or address enabled. Either the local system has not enabled a protocol or an address that it should have, or a remote system is trying to use a protocol that is locally unsupported.

User buffer unavailable *

The message was discarded because there was no user buffer available to receive it. The user has not supplied enough buffers in the user process.

D.8 Physical Link Layer Events *

6.0 Data set ready transition

A transition in the dataset ready signal was detected by the interface hardware.

The message displays the name of the line on which the transition occurred.

6.1 Ring indicator transition

A transition in the ring indicator signal was detected by the interface hardware.

This message displays the name of the line on which the transition occurred.

6.2 Unexpected carrier transition

A transition in the carrier signal was detected by the interface hardware.

This message displays the name of the line on which the transition occurred.

6.3 Memory access error

The interface hardware tried to access nonexistent memory.

This message displays the name of the line on which the error occurred.

6.4 Communications interface error

The interface hardware detected an error in itself.

This message displays the name of the line on which the error occurred.

6.5 Performance error

A soft error has been detected. This error may degrade performance.

This message displays the name of the line on which the error was detected.

D.9 RSX System-specific Events

64.1 Routing data base corrupt

The routing algorithm detected an inconsistency in the routing data bases. The algorithm will not send routing messages.

This message displays the event text only.

64.2 Routing data base restored

The routing data base has been restored by the receipt of valid routing messages from other nodes. The routing algorithm will start sending routing messages again.

This message displays the event text only.

68.14 Normal usage terminated

The operator turned off the circuit.

This message displays the name of the circuit that was shut down.

94.0 DCE detected packet error

The DCE detected an X.25 level 3 error on a circuit.

This message displays the name of the DTE associated with that circuit, the diagnostic code received from the network, and the first 6 bytes of the diagnostic packet.

96.0 State change

This message displays the reason for the X.25 server module state change, the old state, and the new state.

Appendix E
Network Counter Summary

The network software maintains counters for lines, circuits, certain modules, nodes, and the system. In some cases, the counters respond to and reflect network events. (Events are defined in the discussion of event logging in Chapter 2.) In other cases, the counters respond to and reflect normal activities such as messages sent and messages received. Individual counter descriptions indicate whether that counter is incremented when a corresponding event occurs (see Appendix D for a description of event messages). Where possible, the reasons why each of the counters might be incremented are included in the description.

In this appendix, counters are presented in alphabetical order within component groups. If you receive a counter number instead of the usual text when executing commands remotely at an RSX node from a non-RSX node, refer to the lists that correlate counter type numbers with counter text in Appendix F. Then look up the counter description in this appendix for full details.

A counter does not display a value greater than its maximum value. When a counter overflows, it locks on the overflow value until it is zeroed (see the ZERO commands in Chapter 6). Counter displays with an angle bracket (>) indicate that the counter has overflowed. For example, if the maximum value for a counter is 254, its overflow display is >254. Each of the counter descriptions in this appendix includes the maximum value of the counter.

Each category of counters maintains a timing counter (seconds since last zeroed) that is zeroed when its associated counters are zeroed and starts when they start. In this way, the timing counter logs the seconds since its associated counters were zeroed to provide a time frame for them. For example, if you examined the system counters and found that the control buffer allocation failed counter registered 700, you can determine the frequency of the failures by also checking the associated seconds since last zeroed counter.

The counter descriptions in this appendix explain some uses of the counters, but they should not be considered an exhaustive trouble-shooting guide. For detailed information on counters or the software design and algorithms they represent, consult the various architectural specifications.

E.1 Circuit Counters

There are five groups of circuit counters. There are three in the Data Link layer, and one each in the Network Management and Transport layers. The Data Link groups cover DDCMP, Ethernet, and X.25 permanent virtual circuits (PVCs).

E.1.1 Network Management Layer: All Circuits

Seconds since last zeroed

This counter starts when the other circuit counters are zeroed. It increments by 2 every 2 seconds. This counter is zeroed when the other circuit counters are zeroed, so as to provide a time frame for them. The overflow value is 65,534.

E.1.2 Transport Layer: All Circuits

Circuit down

This counter records the number of times that a circuit was declared down by the executor. The overflow value is 254.

Corruption loss

This counter records the number of times that a checksum failure occurred on a PSN. The overflow value is 254.

Initialization failure

This counter increments when the circuit could not be initialized by the executor for network use. The overflow value is 254.

Originating packets sent

This counter records the number of packets sent by the executor over the circuit. The overflow value is 4,294,967,294.

Terminating packets received

This counter records the number of packets received by the executor with the executor as the destination. The overflow value is 4,294,967,294.

Transit congestion loss

This counter records the number of packets received by the executor that were to be routed to another node but were discarded because of heavy traffic on the output circuit. The overflow value is 65,534. This counter is kept for routing nodes only.

Transit packets received

This counter increments when the executor receives a packet that is to be routed to another node. The overflow value is 4,294,967,294. This counter is kept for routing nodes only.

Transit packets sent

This counter increments when the executor sends a packet through to another node. The overflow value is 4,294,967,294. This counter is kept for routing nodes only.

E.1.3 Data Link Layer: DDCMP Circuits

Bytes received

This counter increments when a data byte is received on the circuit. The count does not include Data Link Protocol overhead or bytes retransmitted by the Data Link layer. It can be used with the counter or data blocks received to determine the inbound traffic load on the circuit. The overflow value is 4,294,967,294.

Bytes sent

This counter increments when a data byte is sent on the circuit. The count does not include Data Link Protocol overhead or bytes retransmitted by the Data Link layer. It can be used with the counter or data blocks received to determine the outbound traffic load on the circuit. The overflow value is 4,294,967,294.

Data blocks received

This counter increments when a data block is received on the circuit. The count does not include Data Link Protocol overhead. This counter can be used as a statistical base for evaluating the other Data Link layer counters. The overflow value is 4,294,967,294.

Data blocks sent

This counter increments when a data block is sent on the circuit. The count does not include Data Link Protocol overhead or blocks retransmitted by the Data Link layer. It can be used as a statistical base for evaluating the other Data Link layer counters. The overflow value is 4,294,967,294.

Data errors inbound

This counter indicates the number of incoming data errors on the circuit. It can have any of the following qualifiers: NAKs sent, data field block check error; NAKs sent, reply response. The overflow value is 254.

Data errors outbound

This counter indicates the number of outgoing data errors on the circuit. It can have any of the following qualifiers: NAKs received, header block check error; NAKs received, data field block check error; NAKs received, reply response. The overflow value is 254.

Local buffer errors

This counter increments when a negative acknowledge (NAK) is sent. It can have any of the following qualifiers: NAKs sent, buffer unavailable; NAKs sent, buffer too small. The overflow value is 254.

Local reply timeouts

This counter increments each time that a message is retransmitted because the retry timer for a sent message expired before an ACK was received from the remote node. The overflow value is 254.

Remote buffer errors

This counter increments when a negative acknowledge (NAK) is received. It can have any of the following qualifiers: NAKs received, buffer unavailable; NAKs received, buffer too small. The overflow value is 254.

Remote reply timeouts

This counter increments each time that a message is retransmitted because the retry timer for a sent message expired before an ACK from your node was received at the remote node. The overflow value is 254.

Selection intervals elapsed

This counter records the number of times that the executor turned a circuit around or selected an adjacent node on both half duplex and multipoint circuits. This counter is used as a statistical base for the evaluation of the counter for selection timeouts. The overflow value is 65,534.

Selection timeouts

This counter records the number of times that the executor turned a circuit around or selected an adjacent node on both half duplex and multipoint circuits but the adjacent node failed to respond within the required time. This can be caused by blocks being lost on the circuit in either direction or by too small a value being specified for the executor's response timer. Blocks are usually lost because of a partial, temporary, or total failure of the communications line. This counter can have the following qualifier: no reply to select. The overflow value is 254.

E.1.4 Data Link Layer: Ethernet Circuits

Bytes received

This counter increments when a data byte is received on the circuit. The count does not include Data Link Protocol overhead or bytes retransmitted by the Data Link layer. It can be used with the counter or data blocks received to determine the inbound traffic load on the circuit. The overflow value is 4,294,967,294.

Bytes sent

This counter increments when a data byte is sent on the circuit. The count does not include Data Link Protocol overhead or bytes retransmitted by the Data Link layer. It can be used with the counter or data blocks received to determine the outbound traffic load on the circuit. The overflow value is 4,294,967,294.

Data blocks received

This counter increments when a data block is received on the circuit. The count does not include Data Link Protocol overhead. This counter can be used as a statistical base for evaluating the other Data Link layer counters. The overflow value is 4,294,967,294.

Data blocks sent

This counter increments when a data block is sent on the circuit. The count does not include Data Link Protocol overhead or blocks retransmitted by the Data Link layer. It can be used as a statistical base for evaluating the other Data Link layer counters. The overflow value is 4,294,967,294.

User buffer unavailable

This counter indicates the total number of times that no user buffer was available for an incoming frame that passed all filtering. User buffers are supplied by users on receive requests. The overflow value is 65,534.

E.1.5 Data Link Layer: X.25 Permanent Virtual Circuits (PVCs)

Bytes received

This counter indicates the number of bytes of data received over the circuit since the counter was last zeroed. The overflow value is 4,294,967,294.

Bytes sent

This counter indicates the number of bytes of data sent over the circuit since the counter was last zeroed. The overflow value is 4,294,967,294.

Data blocks received

This counter indicates the number of data blocks received over the circuit since the counter was last zeroed. The overflow value is 4,294,967,294.

Data blocks sent

This counter indicates the number of data blocks sent over the circuit since the counter was last zeroed. The overflow value is 4,294,967,294.

Locally initiated resets

This counter indicates the number of resets sent over the circuit since the counter was last zeroed. The overflow value is 254.

Network initiated resets

This counter increments when a reset is sent by the PSN and is received by the local DTE over this circuit. The overflow value is 254.

Remotely initiated resets

This counter indicates the number of resets received by the local DTE that were originated by a remote DTE or DTEs over this circuit. The overflow value is 254.

E.2 Line Counters

There are five groups of line counters. Four groups are related to the Data Link layer and one group to the Network Management layer. The Data Link layer counters are divided into two groups that are device oriented and two groups that are line oriented.

E.2.1 Data Link Layer: All Devices Except DA, DMC, DMP, PCL, UNA, and QNA

Local process errors

This counter increments when the executor is responsible for a processing fault. On a multipoint line, the counter is kept as a total only for the control station, not for each tributary. Although the fault is usually caused by a programming error at the executor, there is a remote possibility that it could be caused by a line error that was not detected by the Data Link Protocol. This counter can have any of the following qualifiers: NAKs received, header format error; NAKs sent, receive overrun; receive overruns, NAK not sent. The overflow value is 254.

Remote process errors

This counter increments when an adjacent node is responsible for a processing fault. On a multipoint line, the counter is kept as a total only for the control station, not for each tributary. Although the fault is usually caused by a programming error or hardware failure at the adjacent node, there is a remote possibility that it could be caused by a line error that was not detected by the Data Link Protocol. This counter can have any of the following qualifiers: NAKs received, receive overrun; NAKs sent, header format error; selection address errors; streaming tributary. The overflow value is 254.

E.2.2 Data Link Layer: PCL Device

Attempts to become master

This counter increments when an attempt to become the master on the PCL bus fails due to other traffic on the bus. The overflow value is 254.

Device errors

This counter increments when the executor detects faulty hardware operation on its own UNIBUS. This counter can have any of the following qualifiers: interrupt timeout, receiver overflow/UNIBUS timeout, receiver overrun, transmitter overflow/UNIBUS timeout, transmitter underrun. The overflow value is 254.

Process errors

This counter increments when faulty operation is detected by the executor on one of the other nodes on the PCL bus or the executor itself. The overflow value is 254. This counter can have any of the following qualifiers:

- Flag format error (indicates a programming error at some other node)
- Multiplexer address error (indicates that the executor's transmitter is not being allowed a time slice on the bus; usually caused by a hardware installation error)
- Unrecognized receiver error (indicates a hardware error during reception of the flag portion of a message)
- Unrecognized station error (indicates that the executor is missing a tributary address or that some other node is using the wrong tributary address to identify itself)

E.2.3 Network Management Layer: All Lines Except DMC

Seconds since last zeroed

This counter starts when the other line counters are zeroed. It increments by 2 every 2 seconds. This counter is zeroed when the other line counters are zeroed, so as to provide a time frame for them. The overflow value is 65,534.

E.2.4 Data Link Layer: Ethernet Lines

Blocks sent, initially deferred

This counter indicates the total number of times that a frame transmission was deferred on its first transmission attempt. It is used to measure Ethernet contention with no collisions. The overflow value is 4,294,967,294.

Blocks sent, multiple collisions

This counter indicates the total number of times that a frame was successfully transmitted on the third or later attempt, after normal collisions on previous attempts. The overflow value is 4,294,967,294.

Blocks sent, single collision

This counter indicates the total number of times that a frame was successfully transmitted on the second attempt after a normal collision on the first attempt. The overflow value is 4,294,967,294.

Bytes received

This counter increments when a data byte is received on the line. The count does not include Data Link Protocol overhead or bytes retransmitted by the Data Link layer. It can be used with the counter or data blocks received to determine the inbound traffic load on the line. The overflow value is 4,294,967,294.

Bytes sent

This counter increments when a data byte is sent on the line. The count does not include Data Link Protocol overhead or bytes retransmitted by the Data Link layer. It can be used with the counter or data blocks received to determine the outbound traffic load on the line. The overflow value is 4,294,967,294.

Collision detect check failure

This counter indicates the approximate number of times that a collision detect was not sensed after a transmission. The overflow value is 65,534.

Data blocks received

This counter increments when a data block is received on the line. The count does not include Data Link Protocol overhead. It can be used as a statistical base for evaluating the other Data Link layer counters. The overflow value is 4,294,967,294.

Data blocks sent

This counter increments when a data block is sent on the line. The count does not include Data Link Protocol overhead or blocks retransmitted by the Data Link layer. It can be used as a statistical base for evaluating the other Data Link layer counters. The overflow value is 4,294,967,294.

Data overrun

This counter indicates the total number of times that the hardware lost an incoming frame because it was unable to keep up with the data rate. The overflow value is 65,534.

Multicast blocks received

This counter indicates the total number of multicast blocks that have been successfully received. The overflow value is 4,294,967,294.

Multicast bytes received

This counter indicates the total number of multicast data bytes that have been successfully received (including bytes in the Ethernet data field, but not the Ethernet data link headers). The overflow value is 4,294,967,294.

Receive failure

This counter indicates the total number of blocks received with some data error. (The blocks are data frames that passed either physical or multicast address comparison.) For each increment of the counter, one of the following types of failure is recorded: block check error, framing error, frame too long. The overflow value is 65,534.

Send failure

This counter indicates the total number of times that a transmit attempt failed. For each increment of the counter, one of the following types of failure is recorded: carrier check failed, excessive collisions, frame too long, open circuit, remote failure to defer, short circuit. The overflow value is 65,534.

System buffer unavailable

This counter indicates the total number of times that no system buffer was available for an incoming frame. This can be any buffer between the hardware and the user buffers (those supplied on receive requests). The overflow value is 65,534.

Unrecognized frame destination

This counter indicates the number of times that a frame was discarded because there was no enabled portal with the protocol type or multicast address. The count includes frames received for the physical address, broadcast address, or multicast address. The overflow value is 65,534.

E.2.5 Data Link Layer: LAPB Lines

Bytes received

This counter increments when a data byte is received on the line. The count does not include Data Link Protocol overhead or bytes retransmitted by the Data Link layer. It can be used with the counter or data blocks received to determine the inbound traffic load on the line. The overflow value is 4,294,967,294.

Bytes sent

This counter increments when a data byte is sent on the line. The count does not include Data Link Protocol overhead or bytes retransmitted by the Data Link layer. It can be used with the counter or data blocks received to determine the outbound traffic load on the line. The overflow value is 4,294,967,294.

Data blocks received

This counter increments when a data block is received on the line. The count does not include Data Link Protocol overhead. It can be used as a statistical base for evaluating the other Data Link layer counters. The overflow value is 4,294,967,294.

Data blocks sent

This counter increments when a data block is sent on the line. The count does not include Data Link Protocol overhead or blocks retransmitted by the Data Link layer. It can be used as a statistical base for evaluating the other Data Link layer counters. The overflow value is 4,294,967,294.

Data errors inbound

This counter indicates the number of incoming data errors resulting from faults on the channel between the local DTE and the DCE. The counter can have any of the following qualifiers: block too long, block check error, reject sent. The overflow value is 254.

Data errors outbound

This counter indicates the number of outgoing data errors resulting from faults on the channel between the local DTE and the DCE. It can include the following qualifier: reject received. The overflow value is 254.

Local buffer errors

This counter increments when a receive not ready (RNR) packet is sent. It can include the following qualifier: RNR sent, buffer unavailable. The overflow value is 254.

Local process errors

This counter increments when a frame reject error (FRMR) is received over the line or when your system is overloaded. The counter can have any of the following qualifiers: transmit underrun; receive overrun; FRMR received, header format error. Normally, the first two qualifiers indicate that the system is overloaded, while the third indicates that the RSX-11 PSI software is malfunctioning. The overflow value is 254.

Local reply timeouts

This counter increments when the retransmit timer for that line expires. The overflow value is 254.

Remote buffer errors

This counter increments when a receive not ready (RNR) packet is received. It can include the following qualifier: RNR received, buffer unavailable. The overflow value is 254.

Remote process errors

This counter increments when an invalid next expected sequence number (N(R)) and a frame reject error (FRMR) are sent over the line. The counter can have any of the following qualifiers: invalid N(R) received; FRMR sent, header format error. Usually these errors indicate that the DCE is malfunctioning. The overflow value is 254.

Remote reply timeouts

This counter indicates the number of times that a frame with the poll bit set has been received over the line — that is, the number of errors resulting from faults on the line. The overflow value is 254.

E.3 Module Counters

There are two types of module counters: protocol module counters and server module counters. The protocol module is the X.25 protocol module. The server modules are the X.25 server and the X.29 server modules.

E.3.1 X.25 Protocol Module

Bytes received

This counter increments when a byte is received by the local DTE. The overflow value is 4,294,967,294.

Bytes sent

This counter increments when a byte is sent by the local DTE. The overflow value is 4,294,967,294.

Calls received

This counter increments when an incoming call is received by the DTE. The overflow value is 65,534.

Calls sent

This counter increments when an outgoing call is sent. The overflow value is 65,534.

Data blocks received

This counter increments when a data block is received by the local DTE. The overflow value is 4,294,967,294.

Data blocks sent

This counter increments when a data block is sent by the local DTE. The overflow value is 4,294,967,294.

Fast selects received

This counter increments when a call is received with the fast select facility specified. The overflow value is 65,534.

Fast selects sent

This counter increments when a call is sent with the fast select facility specified. The overflow value is 65,534.

Locally initiated resets

This counter increments when a reset is sent by the local DTE. The overflow value is 254.

Maximum channels active

This counter indicates the number of switched virtual circuits that were active at any one time since the counters were last zeroed. These circuits are ones whose logical channel numbers appear in the channel list regardless of whether or not the circuits are used for incoming or outgoing calls. The overflow value is 65,534.

Maximum switched circuits active

This counter indicates the number of switched virtual circuits that were active at any one time since the counters were last zeroed. The overflow value is 65,534.

Network-initiated resets

This counter increments when a reset is sent by the PSN and is received by the local DTE. The overflow value is 254.

Received call resource errors

This counter increments when an incoming call is rejected because of insufficient resources. The overflow value is 65,534.

Remotely initiated resets

This counter indicates the number of resets received by the local DTE that were originated by a remote DTE or DTEs. The overflow value is 254.

Restarts

This counter indicates the number of times that the restart procedure was used on the DTE. The overflow value is 254.

Seconds since last zeroed

This counter starts when the other protocol module counters are zeroed. It increments by 2 every 2 seconds. This counter is zeroed when the other protocol module counters are zeroed, so as to provide a time frame for them. The overflow value is 65,534.

E.3.2 X.25/X29 Server Modules

Incoming calls rejected, no resources

This counter indicates the number of times that the incoming call handler rejected a request to set up a virtual circuit because of insufficient resources. The overflow value is 254.

Logical links rejected, no resources

This counter increments each time that a logical link could not be established because of insufficient resources. The overflow value is 254.

Maximum circuits active

This counter indicates the number of switched virtual circuits that have been set up since the counters were last zeroed. The overflow value is 65,534.

Seconds since last zeroed

This counter starts when the other server module counters are zeroed. It increments by 2 every 2 seconds. This counter is zeroed when the other server module counters are zeroed, so as to provide a time frame for them. The overflow value is 65,534.

E.4 Node Counters

E.4.1 Network Management Layer

Seconds since last zeroed

This counter starts when the other node counters are zeroed. It increments by 2 every 2 seconds. This counter is zeroed when the other node counters are zeroed, so as to provide a time frame for them. The overflow value is 65,534.

E.4.2 Network Services Layer

Bytes received

This counter increments when user data bytes are received from the associated node at the logical link level. It includes only the user data from data messages and from interrupt, connect, accept, reject, disconnect, and abort functions. The overflow value is 4,294,967,294.

Bytes sent

This counter increments when user data bytes are sent to the associated node at the logical link level. It includes only the acknowledged user data from data messages and from interrupt, connect, accept, reject, disconnect, and abort functions. It does not include retransmissions. The overflow value is 4,294,967,294.

Connects received

This counter increments when a connect initiation signal is received from the associated node. The overflow value is 65,534.

Connects sent

This counter increments when a connect initiation signal is sent to the associated node. The overflow value is 65,534.

Messages received

This counter increments when a message is received from the associated node at the logical link level. The count includes both user messages and logical link protocol control messages. Furthermore, it includes internal segmentation of user messages by the Network Services layer. The overflow value is 4,294,967,294.

Messages sent

This counter increments when a message is sent to the associated node at the logical link level. The count includes both user messages and logical link protocol control messages. It also includes the retransmission of a message. The Network Services layer segments the user messages. The overflow value is 4,294,967,294.

Node maximum logical links active

This counter records the number of active links between the executor and the associated node. The overflow value is 65,534.

Received connect resource errors

This counter increments when the executor attempts to reject a connect initiation signal from the associated node due to lack of resources in the executor. This condition occurs when there are not enough maximum links allowed or when there are not enough control buffers or small buffers. The overflow value is 65,534.

Response timeouts

This counter increments when the associated node fails to respond within the required time. This situation can be caused either by messages being discarded in the network (see Section E.1.2) or by a wide variance in the round-trip delay to the node. This condition normally indicates an overload condition in the network. This should be considered a problem if 2 percent or more of the messages sent are timed out. The overflow value is 65,534.

Total maximum logical links active

This counter records the number of active logical links between the executor and all nodes (including itself). This counter is included in counters for the executor node only. The overflow value is 65,534.

Total received connect resource errors

This counter increments when the executor attempts to reject an initiation signal from any node due to the lack of resources in the executor. This counter is included in counters for the executor node. The condition is caused by the maximum logical link parameter being set too low or by the number of buffers parameter set too low for control buffers or small buffers. The overflow value is 65,534.

E.4.3 Executor Node Counters

Aged packet loss

This counter increments when a packet is discarded due to visiting too many nodes. The count is the total of all such discards by the executor node. Only routing nodes keep this counter. This counter is incremented each time the aged packet loss event occurs. The overflow value is 254.

Node out of range packet loss

This counter increments when a packet is discarded because the destination node address was greater than the maximum address defined for the executor. The count is the total of all such discards by the executor node. Only routing nodes keep this counter. This counter is incremented each time the node out of range packet loss event occurs. The overflow value is 254.

Node unreachable packet loss

This counter increments when a packet is discarded because its destination node was unreachable. The count is the total of all such discards by the executor node. Only routing nodes keep this counter. The counter is incremented each time the node unreachable packet loss event occurs. The overflow value is 65,534.

Oversized packet loss

This counter increments when a packet is discarded because it was larger than the circuit buffer size. The circuit buffer size was previously established between the executor node and the adjacent node. Only routing nodes keep this counter. The counter is incremented each time the oversized packet loss event occurs. The overflow value is 254.

Packet format error

This counter increments when a packet is discarded because of invalid packet control information. The count is the total of all such discards by the executor node. This counter is incremented each time the packet format error event occurs. The overflow value is 254.

Partial routing update loss

This counter increments when part of a routing update is lost because it contained a reachable node address that exceeded the maximum address defined for the executor node. The count is the total of all such occurrences at the executor node. Only routing nodes keep this counter. This counter is incremented each time the partial routing update loss event occurs. The overflow value is 254.

Verification reject

This counter increments when the executor rejects a verification request from an adjacent node during routing initialization. The count is the total of all such occurrences at the executor node. This counter is incremented each time the verification reject event occurs. The overflow value is 254.

E.5 System Counters

There are five system counters: four buffer allocation failure counters and one timing counter. Any of the following buffer allocation failures can be resolved using CFE. Use the CFE LIST SYSTEM CHARACTERISTICS command to determine the present setting of the buffer in question. Then, use the CFE DEFINE SYSTEM command to increase the number of those buffers. After the number of buffers is increased, reload DECnet.

Control buffer allocation failed

This counter increments when a control buffer allocation fails. This condition is caused by not having enough control buffers to service the work load on the Communications Executive and its processes. The overflow value is 65,534.

Large buffer allocation failed

This counter increments when a large buffer allocation fails. This condition is caused by not having enough large buffers available to service the work load on the Communications Executive and its processes. It can also be a symptom of the minimum receive buffer level reserving too many large buffers. The overflow value is 65,534.

Receive buffer allocation failed

This counter increments when a receive buffer allocation fails. This condition is caused by not having enough receive buffers reserved from the large buffers to service the work load on the Communications Executive and its processes. The overflow value is 65,534.

Seconds since last zeroed

This counter starts when the other system counters are zeroed. It increments by 2 every 2 seconds. This counter is zeroed when the other system counters are zeroed, so as to provide a time frame for them. The overflow value is 65,534.

Small buffer allocation failed

This counter increments when a small buffer allocation fails. This condition is caused when there are not enough small buffers to service the work load on the Communications Executive and its processes. The overflow value is 65,534.

Appendix F
Network Parameter and Counter Type Numbers

Every parameter or counter affected by the SET, CLEAR, or SHOW commands across the network is assigned a type number. This appendix lists these type numbers as recognized by DECnet-RSX network management software for all component parameters and counters (where applicable). Counter and parameter type numbers from 2300 to 2499 are RSX system specific. Appendix E provides detailed descriptions of network counters. The *DNA Network Management Functional Specification* describes all standard type numbers for all parameters and counters.

This appendix also lists the event types supported for the DECnet-RSX logging component. Appendix D provides detailed descriptions of event classes and types.

Parameter/counter type numbers appear in displays of information from remote nodes that are not able to transmit the text description of a parameter or counter. A non-RSX node running NCP prints the type numbers if it cannot recognize them. For example, if you were on a VMS node, you could set the executor to RSX node BOS and issue the following command:

```
NCP TELL BOS SHOW CIRCUIT PCL-0.0 CHARACTERISTICS
```

The system would then display the characteristics of circuit PCL-0.0:

```
Circuit Volatile Characteristics as of 11-AUG-1983 11:30:34

Circuit = PCL-0.0

Cost                         = 3
Hello timer                  = 15
Listen timer                 = 30
Owner                        = XPT
Type                         = DDCMP control
Tributary                    = 2
```

F.1 Alias Parameters (RSX system specific)

Number	Keywords
100	SCOPE
110	DESTINATION

F.2 Circuit Parameters and Counters

F.2.1 Circuit Parameters

Number	Keywords
0	STATE
1	*substate*
110	COUNTER TIMER
400	LOOPBACK NAME
800	ADJACENT NODE
810	BLOCK SIZE
900	COST
902	ROUTER PRIORITY
906	HELLO TIMER
907	LISTEN TIMER
920	MAXIMUM RECALLS
921	RECALL TIMER
927	MAXIMUM BROADCAST ROUTERS
1100	OWNER
1111	USAGE
1112	TYPE
1120	DTE
1121	CHANNEL
1122	MAXIMUM DATA
1123	MAXIMUM WINDOW
1140	TRIBUTARY
2320	MULTIPOINT ACTIVE (RSX specific)

F.2.2 Circuit Counters

Some counters include qualifiers identified by bit numbers in a bit mask. These qualifiers are included in the following lists, indented under the counter to which they belong.

The following counters are kept for all circuits:

Number	Standard text
0	Seconds since last zeroed
800	Terminating packets received
801	Originating packets sent
805	Corruption loss
810	Transit packets received
811	Transit packets sent
812	Transit congestion loss
820	Circuit down
821	Initialization failure

The following counters are kept for DDCMP circuits:

Number	Standard text
1000	Bytes received
1001	Bytes sent
1010	Data blocks received
1011	Data blocks sent
1020	Data errors inbound
	1 NAKs sent, data field block check error
	2 NAKs sent, REP response
1021	Data errors outbound
	0 NAKs received, header block check error
	1 NAKs received, data field block check error
	2 NAKs received, REP response
1030	Remote reply timeouts
1031	Local reply timeouts
1040	Remote buffer errors
	0 NAKs received, buffer unavailable
	1 NAKs received, buffer too small
1041	Local buffer errors
	0 NAKs sent, buffer unavailable
	1 NAKs sent, buffer too small
1050	Selection intervals elapsed
1051	Selection timeouts
	0 No reply to select

The following counters are kept for Ethernet circuits:

Number	Standard text
1000	Bytes received
1001	Bytes sent
1010	Data blocks received
1011	Data blocks sent
1065	User buffer unavailable

The following counters are kept for permanent X.25 circuits:

Number	Standard text
1000	Bytes received
1001	Bytes sent
1010	Data blocks received
1011	Data blocks sent
1240	Locally initiated resets
1241	Remotely initiated resets
1242	Network-initiated resets

F.3 Line Parameters and Counters

F.3.1 Line Parameters

Number	Keywords	
0	STATE	
1	*substate*	
100	SERVICE	
110	COUNTER TIMER	
1111	DUPLEX	
1112	PROTOCOL	
1151	DEAD TIMER	
1152	DELAY TIMER	
2300	OWNER	(RSX specific)
2310	CONTROLLER CSR	(RSX specific)
2311	UNIT CSR	(RSX specific)
2312	VECTOR	(RSX specific)
2313	PRIORITY	(RSX specific)
2321	MULTIPOINT DEAD	(RSX specific)
2330	LOCATION	(RSX specific)

F.3.2 Line Counters

Some counters include qualifiers identified by bit numbers in a bit mask. These qualifiers are included in the following lists, indented under the counter to which they belong.

The following counters are kept for point-to-point and tributary DDCMP lines:

Number	Standard text
0	Seconds since last zeroed
1100	Remote process errors
	0 NAKs received, receive overrun
	1 NAKs sent, header format error
	2 Selection address errors
	3 Streaming tributary
1101	Local process errors
	0 NAKs sent, receive overrun
	1 Receive overruns, NAK not sent
	3 NAKs received, header format error

DDCMP software does not support the following counters as specified by the network architecture:

- NAKs sent, header block check error: bit 0 for data errors inbound.

  This is counted as a data error, under bit 1, due to information not returned by driver.

- NAKs sent, buffer unavailable: bit 0 for local buffer errors.

  This is not sensed by driver.

- Incomplete reply to select: bit 1 for selection timeouts.

  This is counted as a no reply, under bit 0, due to information not sensed by driver.

- Transmit underruns: bit 2 for local process errors.

  The driver returns a device timeout error, which is then recorded as a no reply to select (bit 0 for selection timeouts) if the line is multipoint or half duplex. Otherwise it is not counted.

The following counters are kept for the PCL device:

Number	Standard text
0	Seconds since last zeroed
2410	Attempts to become master
2411	Process errors
	0 Unrecognized receiver error
	1 Unrecognized station error
	2 Flag format error
	3 Multiplexer address error
2412	Device errors
	0 Transmitter underrun
	1 Transmitter overflow/UNIBUS timeout
	2 Receiver overrun
	3 Receiver overflow/UNIBUS timeout
	4 Interrupt timeout

The following counters are kept for Ethernet lines (UNA and QNA):

Number	Standard text
0	Seconds since last zeroed
1000	Bytes received
1001	Bytes sent
1002	Multicast bytes received
1010	Data blocks received
1011	Data blocks sent
1012	Multicast blocks received
1013	Blocks sent, initially deferred
1014	Blocks sent, single collision
1015	Blocks sent, multiple collisions
1060	Send failure
	0 Excessive collisions
	1 Carrier check failed
	2 Short circuit
	3 Open circuit
	4 Frame too long
	5 Remote failure to defer
1061	Collision detect check failure
1062	Receive failure
	0 Block check error
	1 Framing error
	2 Frame too long
1063	Unrecognized frame destination
1064	Data overrun
1065	System buffer unavailable

The following counters are kept for LAPB lines:

Number	Standard text
0	Seconds since last zeroed
1000	Bytes received
1001	Bytes sent
1010	Data blocks received
1011	Data blocks sent
1020	Data errors inbound
	3 Block too long
	4 Block check error
	5 REJ sent
1021	Data errors outbound
	3 REJ received
1030	Remote reply timeouts
1031	Local reply timeouts
1040	Remote buffer errors
	2 RNR received, buffer unavailable
1041	Local buffer errors
	2 RNR sent, buffer unavailable
1100	Remote process errors
	4 Invalid N(R) received
	5 FRMR sent, header format error
1101	Local process errors
	2 Transmit underrun
	4 Receive overrun
	5 FRMR received, header format error

F.4 Logging Parameters and Events

F.4.1 Logging Parameters

Number	Keywords
0	STATE
100	NAME
200	SINK NODE
201	EVENTS

F.4.2 Logging Events

Most of the following events occur on both routing and nonrouting nodes. Events that occur only on routing nodes are flagged with an asterisk. See Appendix D for descriptions of these events.

Number	Standard text
0.0	Event records lost
0.1	Automatic node counters
0.2	Automatic line counters
0.3	Automatic line service
0.4	Line counters zeroed
0.5	Node counters zeroed
0.6	Passive loopback
0.7	Aborted service request
0.8	Automatic counters
0.9	Counters zeroed

2.0	Local node state change
2.1	Access control failure
3.0	Invalid message
3.1	Invalid flow control
3.2	Node data base reused
4.0 *	Aged packet loss
4.1 *	Node unreachable packet loss
4.2 *	Node out of range packet loss
4.3	Oversized packet loss
4.4	Packet format error
4.5 *	Partial routing update loss
4.6	Verification reject
4.7	Circuit down, circuit fault
4.8	Circuit down, software fault
4.9	Circuit down, operator fault
4.10	Circuit up
4.11	Circuit initialization failure, circuit fault
4.12	Circuit initialization failure, software fault
4.13	Circuit initialization failure, operator fault
4.14	Node reachability change
4.15	Adjacency up
4.16	Adjacency rejected
4.18	Adjacency down
4.19	Adjacency down — operator initiated
5.0	Locally initiated state change
5.1	Remotely initiated state change
5.2	Protocol restart received in maintenance mode
5.3	Send error threshold
5.4	Receive error threshold
5.5	Select error threshold
5.6	Block header format error
5.7	Selection address error
5.8	Streaming tributary
5.9	Local buffer too small
5.10	Restart
5.11	State change
5.12	Retransmit maximum exceeded
5.13	Initialization failure
5.14	Send failed
5.15	Receive failed
6.0	Data set ready transmission
6.1	Ring indicator transmission
6.2	Unexpected carrier transmission
6.3	Memory access error
6.4	Communications interface error
6.5	Performance error
64.1 *	Routing database corrupt
64.2 *	Routing database restored
68.14	Normal usage terminated
94.0	DCE detected packet error
96.0	State change

F.5 Node Parameters and Counters

F.5.1 Node Parameters

Number	Keywords	
0	STATE	
100	IDENTIFICATION	
101	MANAGEMENT VERSION	
114	HARDWARE ADDRESS	
115	SERVICE NODE VERSION	
120	LOAD FILE	
121	SECONDARY LOADER	
122	TERTIARY LOADER	
123	DIAGNOSTIC FILE	
125	SOFTWARE TYPE	
130	DUMP FILE	
135	DUMP ADDRESS	
136	DUMP COUNT	
140	HOST	
141	HOST	
150	LOOP COUNT	
151	LOOP LENGTH	
152	LOOP WITH	
154	LOOP HELP	
500	NAME	
501	CIRCUIT	
502	ADDRESS	
600	ACTIVE LINKS	
601	DELAY	
700	NSP VERSION	
710	MAXIMUM LINKS	
810	TYPE	
820	COST	
821	HOPS	
822	CIRCUIT	
830	NEXT NODE	
900	ROUTING VERSION	
901	TYPE	
910	ROUTING TIMER	
911	SUBADDRESSES	
912	BROADCAST ROUTING TIMER	
920	MAXIMUM ADDRESS	
921	MAXIMUM CIRCUITS	
922	MAXIMUM COST	
923	MAXIMUM HOPS	
924	MAXIMUM VISITS	
926	MAXIMUM BROADCAST NONROUTERS	
932	SEGMENT BUFFER SIZE	
2300	RECEIVE PASSWORD	(RSX specific)
2301	TRANSMIT PASSWORD	(RSX specific)
2310	VERIFICATION STATE	(RSX specific)

F.5.2 Node Counters

Most node counters are kept on both routing and nonrouting nodes. Counters that are kept only on routing nodes are flagged with an asterisk.

Number	Standard text
0	Seconds since last zeroed
600	Bytes received
601	Bytes sent
610	Messages received
611	Messages sent
620	Connects received
621	Connects sent
630	Response timeouts
700	Total maximum logical links active
710	Received connect resource errors
900 *	Aged packet loss
901 *	Node unreachable packet loss
902 *	Node out of range packet loss
903 *	Oversized packet loss
910	Packet format error
920 *	Partial routing update loss
930	Verification reject
2300	Node maximum logical links active
2310	Total received connect resource errors

F.6 Object Parameters (RSX system specific)

Number	Keywords
500	NAME
510	COPIES
511	USER
520	VERIFICATION

F.7 Process Parameters (RSX system specific)

Number	Keywords
0	STATE
10	LOCATION
20	MAXIMUM LINES
21	MAXIMUM CONTROLLERS
30	PARTITION

F.8 System Parameters and Counters (RSX system specific)

F.8.1 System Parameters

Number	Keywords
10	ACTIVE CONTROL BUFFERS
20	ACTIVE SMALL BUFFERS
30	ACTIVE LARGE BUFFERS
110	MAXIMUM CONTROL BUFFERS
120	MAXIMUM SMALL BUFFERS
130	MAXIMUM LARGE BUFFERS
131	LARGE BUFFER SIZE
140	MINIMUM RECEIVE BUFFERS

F.8.2 System Counters

Number	Standard text
0	Seconds since last zeroed
10	Control buffer allocation failed
20	Small buffer allocation failed
30	Large buffer allocation failed
40	Receive buffer allocation failed

F.9 X.25 Access Module Parameters

Number	Keywords
2310	DESTINATION
2320	NUMBER
2330	SCOPE

F.10 X.25 Protocol Module Parameters and Counters

F.10.1 X.25 Protocol Module Parameters

Number	Keywords
0	STATE
100	COUNTER TIMER
1100	DTE
1101	GROUP
1110	NETWORK
1120	LINE
1140	DEFAULT DATA
1141	DEFAULT WINDOW
1150	MAXIMUM DATA
1151	MAXIMUM WINDOW
1152	MAXIMUM CLEARS
1153	MAXIMUM RESETS
1154	MAXIMUM RESTARTS
1160	CALL TIMER
1161	CLEAR TIMER
1162	RESET TIMER
1163	RESTART TIMER
1170	DTE (qualified by GROUP)
1171	NUMBER
1172	TYPE

F.10.2 X.25 Protocol Module Counters

Number	Standard text
0	Seconds since last zeroed
1000	Bytes received
1001	Bytes sent
1010	Data blocks received
1011	Data blocks sent
1200	Calls received
1201	Calls sent
1210	Fast selects received
1211	Fast selects sent
1220	Maximum switched circuits active
1221	Maximum channels active
1230	Received call resource errors
1240	Locally initiated resets
1241	Remotely initiated resets
1242	Network-initiated resets
1250	Restarts

F.11 X.25/X.29 Server Module Parameters and Counters

F.11.1 X.25/X.29 Server Module Parameters

Number	Keywords
0	STATE
100	COUNTER TIMER
300	DESTINATION
310	MAXIMUM CIRCUITS
340	OBJECT
350	PRIORITY
351	CALL MASK
352	CALL VALUE
353	GROUP
354	NUMBER
355	SUBADDRESSES

F.11.2 X.25/X.29 Server Module Counters

Number	Standard text
0	Seconds since last zeroed
200	Maximum circuits active
210	Incoming calls rejected, no resources
211	Logical links rejected, no resources

INDEX

E

F

G

M

N

O

P

Q

R

S

T

U

V

W

X

Z

DECnet-RSX
System Manager's Guide
AA-H224C-TC

READER'S COMMENTS

NOTE: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. If you require a written reply and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Did you find this manual understandable, usable, and well-organized? Please make suggestions for improvement.

Did you find errors in this manual? If so, specify the error and the page number.

Please indicate the type of user/reader that you most nearly represent.

- ☐ Assembly language programmer
- ☐ Higher-level language programmer
- ☐ Occasional programmer (experienced)
- ☐ User with little programming experience
- ☐ Student programmer
- ☐ Other (please specify) ____________

Name ____________ Date ____________

Organization ____________

Street ____________

City ____________ State ____________ Zip Code or Country ____________

—Do Not Tear - Fold Here and Tape—

digital

No Postage
Necessary
if Mailed in the
United States

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO.33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

SOFTWARE DOCUMENTATION
1925 ANDOVER STREET TW/E07
TEWKSBURY, MASSACHUSETTS 01876

Do Not Tear - Fold Here and Tape

Cut Along Dotted Line